

E.T.S. de Ingeniería Industrial,
Informática y de Telecomunicación

Aplicación de herramientas de Business Intelligence en datos del entorno de Salud



Grado en Ingeniería Informática

Trabajo Fin de Grado

Uxue Ayechu Abendaño, autora

José Javier Astrain Escola, director

Pamplona, 23 de junio de 2017

Agradecimientos

En primer lugar, quisiera expresar mi agradecimiento a la Universidad Pública de Navarra y en especial a la Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación donde he recibido durante estos años, no sólo una formación académica, sino también una formación humana.

A mi tutor de Trabajo Fin de Grado, José Javier Astrain Escola. Gracias por todo el esfuerzo que dedicas a tus alumnos y a la Escuela, has sabido ayudarme a organizar mi conocimiento y plasmarlo en este trabajo.

También agradecer a Gobierno de Navarra la oportunidad de trabajar con ellos, donde he aprendido muchísimo y conocido de primera mano la exigencia del mundo laboral. Este gran y ambicioso proyecto, me ha ayudado a desarrollarme profesionalmente en el ámbito de Business Intelligence, Además los datos del entorno de Salud, quizás los más peculiares, cumplen todas las anomalías posibles, lo cual me ha hecho aprender a afrontar diversos problemas con ingeniosas soluciones. Pero este proyecto no acaba aquí, sino que no ha hecho más que empezar.

A mi familia, en especial a mis padres y hermanos, siendo siempre un apoyo para mí.

A mis compañeros y amigos de promoción, por acompañarme durante estos años.

Resumen

El proyecto surge de la necesidad de explotar la información sanitaria de Navarra. Anteriormente, esta necesidad se intentaba satisfacer mediante la creación de listas impresas, hojas de cálculo, Access vinculados a bases de datos OLTP, cuadros de mando en PDF, etc. Actualmente, estos métodos son insostenibles debido al alto coste de personal técnico que requiere la creación de los informes manualmente y a las dificultades para explotar los distintos productos con información clínica de manera conjunta.

Para ello se ha desarrollado un sistema de Business Intelligence (BI) que integra la información clínica y administrativa proveniente de los diversos productos como Atenea, Lakora, Leire, HCI y Lamia. El sistema provee un Data Mart con varias dimensiones y tablas de hechos muy interesantes para el estudio de los datos sanitarios. Dichas dimensiones y tablas integran la información de todos los productos mencionados obteniendo información completa y de calidad. Además del Data Mart, se generan informes muy valiosos, como por ejemplo el cuadro de mandos de crónicos, que muestra indicadores sobre los pacientes crónicos de Navarra, con varias agregaciones, por zona y por plaza.

El Data Mart y los cuadros de mandos se emplean para ayudar a la toma de decisiones importantes como el destino de los recursos de sanidad de Gobierno de Navarra.

Palabras clave

Business intelligence, Data warehouse, Data mart, ETL, Tableau

Abstract

This project arises from the necessity to exploit the health information of Navarre. Previously, this necessity was intended to be met by creating printed lists, spreadsheets, Access linked to OLTP databases, PDF dashboards, and so on. Currently, these methods are unsustainable due to the high cost of technical personnel that requires the creation of manual reports and the difficulties to exploit the different products with clinical information jointly.

For this reason, it is proposed the development of a Business Intelligence (BI) system that integrates clinical and administrative information from the products such as Atenea, Lakora, Leire, HCI and Lamia. In addition, the system provide a Data Mart with dimensions and tables of facts very interesting for health data analysis. Those dimensions and tables, generate complete and quality information. Furthermore, we generate valuable dashboards, for example a control panel for chronic patients that shows indicators about chronic patients in Navarre, with different aggregations, by zone and place.

This panel controls and Data Mart helps take important decisions such as the destination of health resources.

Keywords

Business intelligence, Data warehouse, Data mart, ETL, Tableau

Índice

1.	Introducción	10
1.1	Introducción	10
1.2	Antecedentes	12
1.3	Objetivos	14
1.4	Estado del arte	14
1.5	Propuesta	17
2.	Conceptos básicos	19
3.	Análisis.....	24
3.1	Situación actual (antes de la realización del proyecto).....	24
3.2	Análisis DAFO	25
3.3	Objetivos principales de negocio	27
3.4	Resultados análisis	28
4.	Diseño.....	30
4.1	Modelo normalizado vs modelo dimensional	30
4.2	Arquitectura de los Sistemas de Explotación.....	33
4.2.1	Sistemas operacionales	33
4.2.2	Sistemas de Explotación - Backend	34
4.2.3	Sistema de Explotación – <i>Frontend</i>	36
5.	Metodología empleada	39
5.1	Scrum.....	39
5.2	Reuniones.....	40
5.2.1	Reunión con el cliente	40
5.2.2	Reunión de planificación	40
5.2.3	Reunión diaria (<i>daily</i>)	41
5.2.4	Reunión review.....	42
5.2.5	Reunión retrospectiva	42
5.3	Análisis de la metodología	43
6.	Desarrollo	44
6.1	Componentes de la solución	44
6.1.1	Interfaces del SE	44
6.1.2	Elementos de puesta en marcha del sistema.....	45
6.1.3	Planificación de puesta en marcha del sistema	46
6.2	Realización del desarrollo	46
6.2.1	Bases de datos de las réplicas	47

6.2.2	Bases de Datos del modelo de BI	50
6.2.3	Bases de Datos de configuración y estado del modelo BI.....	67
7.	Funcionamiento	74
8.	Conclusiones y líneas futuras	76
9.	Bibliografía y referencias.....	78
10.	Anexos.....	79
10.1	Requerimientos del cliente, solicitudes	79
10.1.1	Sprint 1	79
10.1.2	Sprint 2	81
10.1.3	Sprint 3	83
10.1.4	Sprint 4	86
10.1.5	Sprint 5	88
10.1.6	Sprint 6	91
10.2	Planificación de cada sprint.....	93
10.2.1	Sprint 1	93
10.2.1	Sprint 2	94
10.2.2	Sprint 3	95
10.2.3	Sprint 4	97
10.2.4	Sprint 5	97
10.3	Procedimiento para el cálculo del indicador número de pacientes polimedicados ...	98

Índice de ilustraciones

Ilustración 1. Esquema organizativo de Gobierno de Navarra	12
Ilustración 2. Esquema de los productos del ámbito de salud soportados por la DGITIP.	13
Ilustración 3 Cuadrado de Gartner comparando herramientas de BI.	15
Ilustración 4 Informe herramienta Power BI	16
Ilustración 5 Interfaz de la herramienta Tableau.....	17
Ilustración 6 Esquema en estrella.	21
Ilustración 7 Tabla Análisis DAFO	27
Ilustración 8 Arquitectura de Kimball	31
Ilustración 9 Arquitectura de Inmon	32
Ilustración 10 Arquitectura de los sistemas de explotación	33
Ilustración 11 Arquitectura de los sistemas de explotación, Backend.....	34
Ilustración 12 Arquitectura de los sistemas de explotación, Frontend	36
Ilustración 13 Fotografía de la pizarra Scrum	42
Ilustración 14 Tabla de mejoras, reunión retrospectiva	43
Ilustración 15 Cuadro de Mando en Tableau de SNS-O (Datos ficticios de 2100)	45
Ilustración 16 Estado de Ejecución.....	46
Ilustración 17 BBDD de las réplicas	47
Ilustración 18 Esquema de los productos replicados.....	48
Ilustración 19 Bases de Datos modelo BI	50
Ilustración 20 Template en Visual Studio para la Extracción de datos	51
Ilustración 21 Data Flow del proceso de Extracción de datos	52
Ilustración 22 Ejemplo tabla extraída de Lakora, con nombres poco representativos	52
Ilustración 23 Extracción de datos. Consulta SQL1_ORIG.....	53
Ilustración 24 Extracción de datos. Normalización y transposición.....	53
Ilustración 25 Extracción de datos. Limpieza	54
Ilustración 26 Extracción de datos. Add Metadata	55
Ilustración 27 Extracción de datos. Tabla1_DEST	55
Ilustración 28 Template en Visual Studio para la Validación de datos	56
Ilustración 29 Data Flow del proceso de Validación de datos	57
Ilustración 30 Template en Visual Studio para la Transformación de datos	58
Ilustración 31 Data Flow del proceso de Transformación de datos.....	59
Ilustración 32 Transformación de datos. Add Metadata	59
Ilustración 33 Transformación de datos. Tabla destino (TF_Centros)	60
Ilustración 34 Template en Visual Studio para el paso al Data Mart de una dimensión	61
Ilustración 35 Data Flow del proceso de paso al Data Mart de una dimensión	62
Ilustración 36 Tabla destino del proceso de paso al Data Mart de una dimensión.....	63
Ilustración 37 Dimensión de centros.....	63
Ilustración 38 Template en Visual Studio para el paso al Data Mart de una tabla de hechos ...	64
Ilustración 39 Data Flow TFDM del proceso de paso al Data Mart de una tabla de hechos	65
Ilustración 40 Data Flow TH del proceso de paso al Data Mart de una tabla de hechos	66
Ilustración 41 Base de datos RIN_conf.....	67
Ilustración 42 Datos de la tabla Dtsx.....	67
Ilustración 43 Datos de la tabla Dtsx_Orig.....	68
Ilustración 44 Datos de la tabla Dtsx_Dest	68
Ilustración 45 Datos tabla Conexiones_OleDB.....	68

Ilustración 46 Contenedor configurable (Orígenes y Destinos) del template de Extracción.....	69
Ilustración 47 Base de datos RIN_conf. Procedimientos almacenados	69
Ilustración 48 Bucle que itera sobre las bases de datos del template de Extracción	70
Ilustración 49 Base de datos RIN_log.....	70
Ilustración 50 Datos tabla Estadísticas.....	71
Ilustración 51 Datos tabla Estado_Ejecucion	71
Ilustración 52 Base de datos RIN_log.....	72
Ilustración 53 Contenedor inicial y final del template de Extracción	72
Ilustración 54 Event Handlers del template de Extracción	73
Ilustración 55 Contenedor penúltimo del template de Extracción.....	73
Ilustración 56 Cuadro de Mandos Crónicos	75

1. Introducción

El proyecto surge de la necesidad de explotar la información sanitaria de Navarra. Anteriormente, esta necesidad se intentaba satisfacer mediante la creación de listas impresas, hojas de cálculo, Access vinculados a bases de datos OLTP, cuadros de mando en PDF, etc. Actualmente, estos métodos son insostenibles debido al alto coste de personal técnico que requiere la creación de los informes manualmente y a las dificultades para explotar los distintos productos con información clínica de manera conjunta. Para ello se propone el desarrollo de un sistema de *Business Intelligence* (BI) que integre información clínica y administrativa proveniente de los diversos productos existentes. Además, contendrá un modelo de información consultable por usuarios analistas, un conjunto de informes orientados a gestores (vistas de Farmacia y de Salud Mental) y varios cuadros de mandos y paneles de control para usuarios clínicos.

1.1 Introducción

Este trabajo fin de grado trata sobre la inteligencia empresarial, más conocida por su nombre en inglés *Business Intelligence* cuyo acrónimo es BI. Su objetivo principal es facilitar la toma de decisiones de la empresa basándose en el análisis de datos. Para realizar dicho análisis se emplean un conjunto de estrategias, aplicaciones, datos de distintas fuentes, productos, tecnologías, etc. Gracias a todas estas herramientas conseguimos medir el rendimiento o funcionamiento del negocio y mostrarlo a las personas indicadas.

En concreto, este estudio se centra en la aplicación de BI al ámbito sanitario. En comparación con su aplicación clásica, a una empresa, este es bastante más complejo, ya que tenemos muchos más parámetros que evaluar para la toma de decisiones, por ejemplo: reducir las listas de espera de los pacientes, intentar aprovechar cada consulta al máximo, optimizar el uso de las máquinas (TAC, hemodiálisis, bombas de infusión...), reducir los costes de los tratamientos sin que estos dejen de ser efectivos, repartir los recursos entre todos los centros según la necesidad, derivar pacientes de unos centros a otros... En general, se trata de optimizar el funcionamiento del servicio sanitario.

Lo primero que hacemos en cualquier proyecto de BI es estudiar los datos disponibles para nuestro modelo. En el ámbito sanitario normalmente se almacenan datos antiguos, por ejemplo, los primeros datos médicos informatizados en EEUU datan de 1950. Evidentemente, estos primeros datos eran más escuetos de lo que son ahora y tampoco seguían la estructura que siguen los datos introducidos actualmente. Estos primeros datos a los que nos referimos se obtienen desde ficheros. Actualmente, la mayoría de los datos que toma el servicio sanitario, se realizan a partir de aplicaciones que utiliza el personal sanitario para registrar su trabajo y las asistencias que realizan. Estos datos se obtienen desde las bases de datos de las distintas aplicaciones que utiliza el Complejo Hospitalario de Navarra. Además, actualmente existen algunos estándares internacionales tales como la Clasificación Internacional y Estadística de Enfermedades y Problemas Relacionados con la Salud (CIE) [1] para la clasificación de enfermedades, Clasificación Internacional de Atención Primaria (CIAP) [2] para la clasificación de atención primaria en hospitales o SNOMED CT (*Systematized Nomenclature*

of Medicine – Clinical Terms) [3]. Estos estándares ayudan a la unificación de los datos sanitarios a nivel internacional y nacional. Además, existen otros estándares que permiten la interoperabilidad y el intercambio electrónico de información médica entre distintos sistemas sanitarios, como es el caso de HL7 (*Health Level Seven*) [4] y DICOM (*Digital Imaging and Communication in Medicine*) [5]. La sanidad navarra también dispone de varias webs en las que también se introducen datos y que posteriormente el equipo de BI obtiene a partir de servicios web. Como se puede observar, el hecho de tener tantas aplicaciones y tan diversas (cada aplicación tiene su propia forma de almacenar los datos) supone un gran esfuerzo unificar todos estos datos en una sola base de datos como pretendemos utilizando BI. Además de todos estos datos que nos proporcionan otros soportes informáticos, también recibimos informes en formato Excel que generan los médicos sobre objetivos, o resultados, que también habrá que añadir a la base de datos de modelo BI.

A continuación aplicaremos la técnica ETL (*Extract, Transform and Load*). Pero para ello previamente debemos realizar un análisis de los datos que vamos a tratar y adquirir cierto conocimiento sobre la empresa (funcionamiento y modelo de negocio), en nuestro caso sobre los hospitales de Navarra. Adquiridos estos conocimientos, pasamos al empleo de la técnica ETL que lo primero que hace es extraer los datos desde las distintas fuentes, a continuación los transforma, y por último los carga en una base de datos.

Lo más pesado del proceso de extracción es comprender los datos de las distintas fuentes. En unas fuentes tendremos que entender los modelos de bases de datos que siguen, en otras los ficheros de datos, los documentos de Access... Además, hay que conocer las relaciones entre las distintas fuentes, si es que las hay, o cómo se denomina en las distintas fuentes a campos comunes. Por ejemplo, el DNI en una fuente puede llamarse NIF, en otra DNI y en otra Id. Esto sucede muy a menudo ya que cada fuente ha trabajado de forma independiente de las demás. Además, en sanidad hay un importante factor y es la necesidad de que todo funcione siempre. Por ejemplo en un servicio de urgencias no puede dejar de funcionar la base de datos de la cual se extraerá información sobre el paciente al que vamos a atender, pues puede ser alérgico a determinados medicamentos o tener una enfermedad importante, datos que el médico necesita conocer. Debido a este motivo no se puede cambiar una aplicación o un sistema informático por otro, sino que únicamente se puede actualizar y tiene que mantener todos los servicios anteriores y los datos. Luego, aunque se creen actualizaciones con mejoras, resulta muy difícil que en estas actualizaciones se tenga en cuenta a las demás fuentes para emplear terminologías consistentes. Teniendo en cuenta que los primeros datos informáticos de sanidad de Navarra no se diseñaron por ningún experto en informática es comprensible que la estructura que siguen los datos no sea la más adecuada. Justo para estos problemas de “integración entre datos de distintas fuentes” trabajamos con herramientas de BI.

En el siguiente paso, el de transformación, lo que se hace es unificar los datos de las distintas fuentes para que sean homogéneos. Esto significa que, por ejemplo, en una base de datos puedo tener un campo que represente el sexo de un cliente y los valores que toma son “M”: Mujer y “H”: Hombre, mientras que en otra base de datos los valores que toma otro campo que representa el sexo son “F”: Femenino y “M”: Masculino; evidentemente, si estas fuentes se mezclan sin realizar un tratamiento previo perderemos mucha información ya que cuando tengamos una M no sabremos si significa Masculino o Mujer. El paso de la transformación se realizará de una forma u otra dependiendo de las fuentes y de lo diversas que sean. Puede

llegar a ser un proceso bastante complejo si las fuentes utilizan distintos identificadores para determinar de qué paciente se trata (por DNI, tarjeta sanitaria...).

En el último paso “Load” se cargan los datos ya transformados en una nueva base de datos que integra todas las fuentes, para poder extraer información sobre el comportamiento total de la empresa. La parte compleja de este paso es el diseño de la nueva base de datos.

Ahora ya solo queda explotar los datos, que hemos unificado en la nueva y única data base, al máximo. Para ello se crean una serie de informes acordes a lo que necesite la empresa u organización en la que se esté trabajando. Estos informes se suelen generar usando CMI (Cuadro de Mandos Integral) que permiten mostrar varios indicadores y objetivos de la empresa desde distintas perspectivas.

1.2 Antecedentes

Este proyecto de BI se ha realizado en colaboración con el Gobierno de Navarra, y concretamente, con la Dirección General de Informática, Telecomunicaciones e Innovación Pública (DGITIP), que se dedica a dar soporte a todas las necesidades informáticas del Gobierno de Navarra, y con el grupo informático específico para el Área de Salud, el Servicio de Sistemas de Información del Área Sanitaria (SSIAS). Dentro del Departamento de Salud, tenemos la Dirección General de Salud, el Servicio Navarro de Salud, el Instituto de Salud Pública y Laboral de Navarra y la Fundación Miguel Servet. La Ilustración 1 muestra el esquema organizativo del Gobierno de Navarra referente al ámbito sanitario.

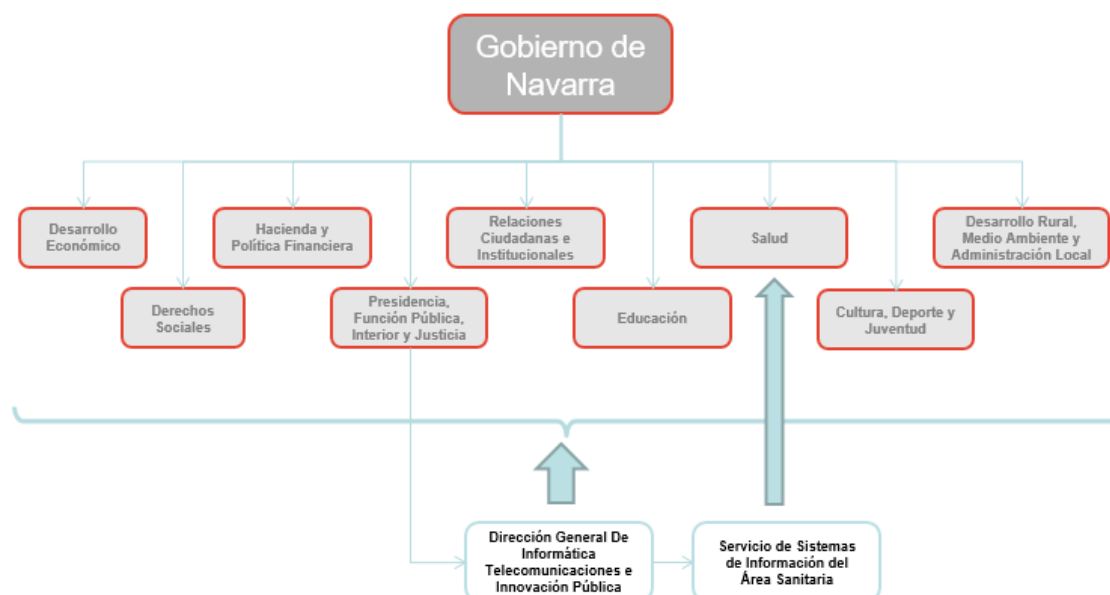


Ilustración 1. Esquema organizativo de Gobierno de Navarra

Las necesidades informáticas de todos ellos se resuelven con aproximadamente 700 productos. El concepto de producto se refiere a un conjunto de elementos informáticos que

interactúan entre sí con el fin de apoyar las actividades de negocio de los usuarios que emplearan el producto. Estos usuarios son normalmente equipos de médicos y personal administrativo de salud. Estos colectivos están representados por el jefe de negocio, quien especifica los requerimientos que su colectivo necesita resolver mediante el empleo de sistemas informáticos. Los productos mencionados se gestionan desde la DGITIP y son productos muy heterogéneos en cuanto a tipo, tamaño, volumen de usuarios y funcionalidades se refiere. La mayoría corresponden a sistemas operacionales OLTP (OnLine Transaction Processing o procesamiento de transacciones en línea), por ejemplo, aplicaciones de gestión administrativa, clínica, etc. La ilustración 2 muestra algunos de estos productos.

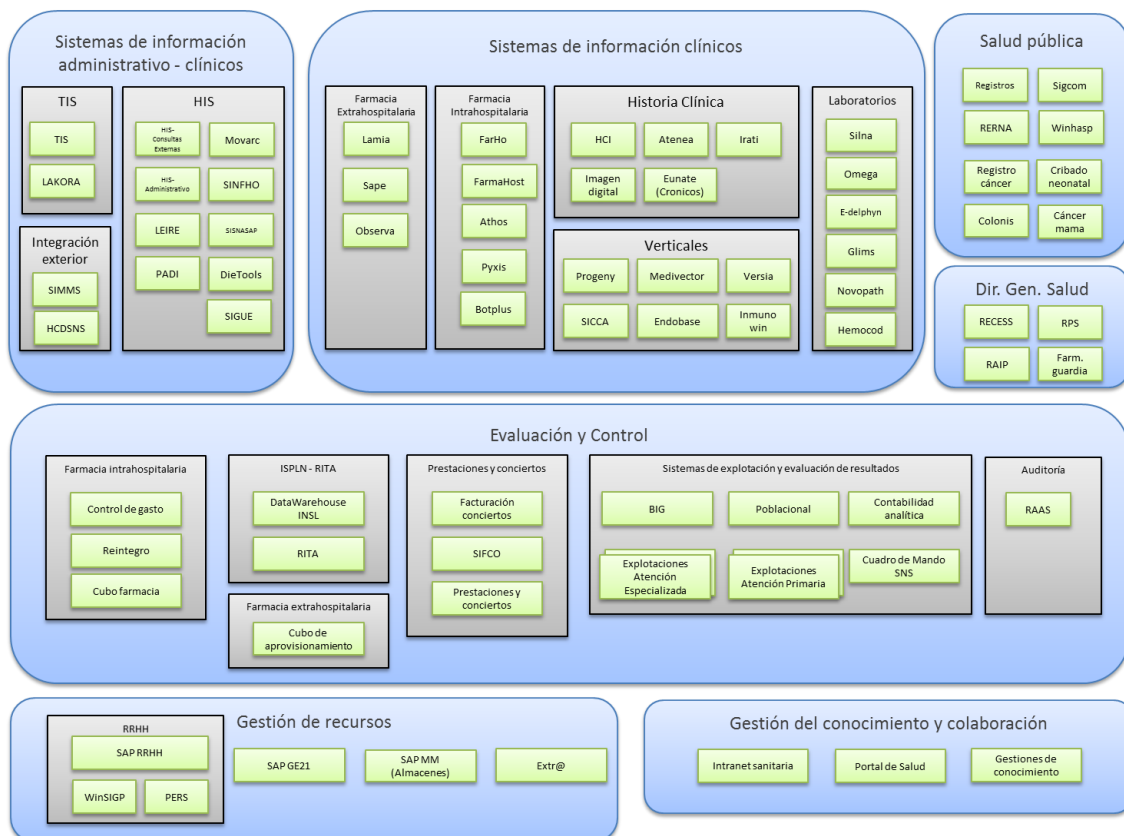


Ilustración 2. Esquema de los productos del ámbito de salud soportados por la DGITIP.

Como se puede pensar, el tener tantos productos supone tener muchos datos que interesa explotar de manera conjunta para obtener información. A lo largo del tiempo, para dar respuesta a estas necesidades se generaban listados impresos o por pantalla, hojas de cálculo de generación manual y automática, informes Access vinculados a bases de datos OLTP, procesos de agregación de datos, cuadros de mando en PDF y unas primeras iniciativas de cubos OLAP (On-Line Analytical Processing). Como se puede intuir, estos métodos iniciales para explotar la información son muy limitados y a comienzos de 2014 el Servicio de Salud solicita información de gestión en todos los niveles de la organización, esto supone tres productos nuevos. El primero es un cuadro de mando navegable para el SNS-O (Servicio Navarro de Salud), una base de datos poblacional que reúna información de varios de los productos mostrados en la Ilustración 1 con información clínica y administrativa de los pacientes e implantar contabilidad analítica en el SNS-O con coste a paciente. Ante esta situación se dan los primeros pasos para poder abordar las solicitudes, y estos son diseñar una arquitectura de sistemas de explotación, evaluar las herramientas para la creación de informes

(Microsoft, Information Builders, Tableau...) y crear una estructura organizativa en la DGITIP (Dirección General De Informática Telecomunicaciones e Innovación Pública) para llevar a cabo este trabajo. Aquí es donde entra mi participación, para este nuevo proyecto, conformado por tres productos, se estimó que se necesitaban un jefe de producto, un gestor de producto y un grupo de 6 desarrolladores. Yo formo parte del equipo de desarrolladores, que se divide en dos grupos, un desarrollador se dedica exclusivamente al producto de contabilidad analítica y los otros 5 nos dedicamos a los otros dos productos, principalmente y al que dedicamos más tiempo es a la base de datos poblacional mencionada anteriormente.

1.3 Objetivos

El principal objetivo del empleo de BI sobre los datos sanitarios de Navarra es extraer información útil y de calidad. Pero se dispone de múltiples fuentes de datos, de génesis y naturaleza muy dispar, y es necesario realizar un tratamiento previo para poder explotar la información disponible, y es aquí donde nace la necesidad de BI. Los datos con los que trabajamos son muy valiosos, pero hay que saber tratarlos y actualmente no existen unos objetivos fijados de antemano sino que en general se quiere poder tener información de todo el sistema sanitario desde los costes hasta los tratamientos. Existen múltiples objetivos y a medida que se emplea la base de datos de BI van aumentando. Entre los objetivos iniciales están el poder controlar los costes de los tratamientos, el disponer de las necesidades de los distintos centros de salud, anticiparse a los brotes de enfermedades... Actualmente la base de datos está “naciendo” y ni los datos son perfectos ni estos datos se explotan todo lo que se podría, queda mucho camino por recorrer. Los principales usuarios de la base de datos de BI son los médicos, quienes no veían los informes que les preparamos como una necesidad, sin embargo a medida que los emplean van viendo que les pueden ser muy útiles y ayudarles en su día a día. Por este motivo cada vez más se están involucrando más en el modelo ayudándonos a constatar los datos y sacar la máxima información pidiendo nuevos cuadros de mando e indicadores. Luego entre los objetivos también está el poder crear un modelo que se pueda explotar de muy diversas maneras y que resulte útil, para ello hay que estar en constante contacto con los usuarios para conocer sus necesidades y poder satisfacerlas.

1.4 Estado del arte

En la introducción ya se ha explicado de forma general el empleo de BI y su utilidad. A continuación se mencionan algunas herramientas que existen actualmente en el mercado para la aplicación del BI y algunos ejemplos de su empleo en distintos entornos, empresas y entornos sanitarios como en el que me centro en este trabajo.

A continuación comentaré algunas de las más conocidas herramientas de BI. Existen dos tipos de productos para realizar BI. Por una parte hay empresas que ofrecen un servicio completo de BI, y otras que ofrecen una herramienta para la base de datos y otra herramienta para la creación de los informes a partir de estas bases de datos. La mayoría de los que ofrecen las dos herramientas, suelen ser compatibles con otras, es decir, puedo emplear una base de datos de Oracle y utilizar una herramienta de Microsoft para generar los informes. En cuanto a los productos que ofrecen la solución completa de BI, suponen un alto coste de tiempo y de

dinero. En este caso no se subcontrató a una empresa para que realizara todo este desarrollo, pues esto conlleva un mínimo de varios años para comenzar a ver resultados del producto, mientras que generando una solución interna, se pueden ofrecer resultados parciales en intervalos menores de tiempo. Por este principal motivo se realizó un equipo de BI para Salud, que es el encargado de generar todo el sistema que explico en este proyecto. Este equipo de BI, emplea herramientas de BI para la generación de informes. A continuación analizamos estas herramientas basándonos en el cuadro mágico de Gartner, todo un referente en el mundo de BI [10].



Ilustración 3 Cuadrado de Gartner comparando herramientas de BI.

Como podemos observar en la Ilustración 3, las dos herramientas mejor valoradas son Tableau, que es la que nosotros empleamos y la solución que ofrece Microsoft. A continuación explico estas dos herramientas que son las más importantes y las que más peso tienen en el mercado. Este cuadrante ha cambiado mucho respecto al del año pasado, cuando Tableau estaba en cabeza, seguido de Qlik y Microsoft. Qlik ha bajado considerablemente, y Tableau y Microsoft se han disparado y están muy distantes de las demás herramientas. Como se apreciaba en la ilustración anterior.

Power BI es la herramienta que ofrece Microsoft junto con SQL Server para el desarrollo de Inteligencia empresarial. La base de datos se crea en SQL Server y los datos se explotan y analizan desde Power BI. Power BI ofrece varios productos Power BI Mobile, Power BI Desktop y Power BI Embedded, cada uno para cubrir distintas necesidades. En nuestro caso de estudio, la herramienta que más se adecua a nuestras necesidades sería Power BI Desktop, ya que por ejemplo la Mobile sirve para teléfonos móviles, mientras que la mayoría de los informes de Salud, se ven a través de ordenadores. Además pocos empleados tienen móvil propio de la empresa, donde instalar la aplicación, de modo que no es la herramienta que más nos convenga. La herramienta mencionada sirve para la creación de informes y visualizaciones. Permite generar los informes a la carta, es decir, que el propio usuario puede modificar los datos que quiere ver, estos informes pueden ser interactivos y también permiten explorar por regiones. Pero para generar todos estos informes se necesita previamente una base de datos sobre la cual realizaremos los cálculos, el sistema de bases de datos que proponen es su propio sistema SQL Server. A continuación, en la ilustración 4, se muestra un informe de ejemplo obtenido con la herramienta Power BI.

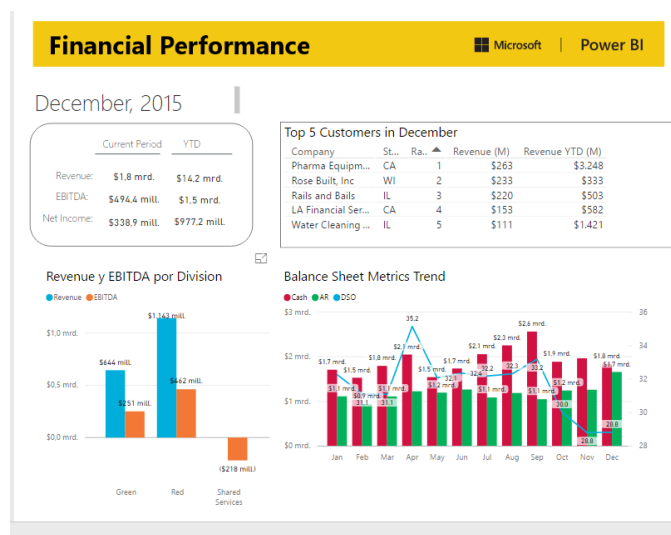


Ilustración 4 Informe herramienta Power BI

Tableau es un software de Business Intelligence, que sirve para facilitar el análisis visual de los datos. Es una herramienta que ayuda a mejorar y acelera la toma de decisiones de las personas encargadas de tomarlas. Sus mayores ventajas, son que dispone de acceso instantáneo a los datos, es independiente de los orígenes de datos que tenga la empresa (permite combinar varias bases de datos en una misma hoja, informe). Permite al usuario gestionar y distribuir sus análisis, el departamento de informática genera un informe general y luego cada usuario puede modificarlo a su gusto. Al igual que Power BI, tiene diferentes productos Tableau Desktop y Tableau Server. En este caso nos interesa Tableau Desktop, que es muy fácil de manejar, para crear informes limpios y sencillos. Es capaz de combinar varios análisis en un solo informe, también elimina el paso previo para crear universos o cubos, pues su conexión es directa a la base de datos. Además en la última versión la 10.3, han añadido la funcionalidad de las alertas, permite generar alertas. Dichas alertas se generan, si un dato supera un rango que le introduzcamos, por ejemplo.

En la ilustración 5 observamos la interfaz de la herramienta durante la creación de un cuadro de mandos.

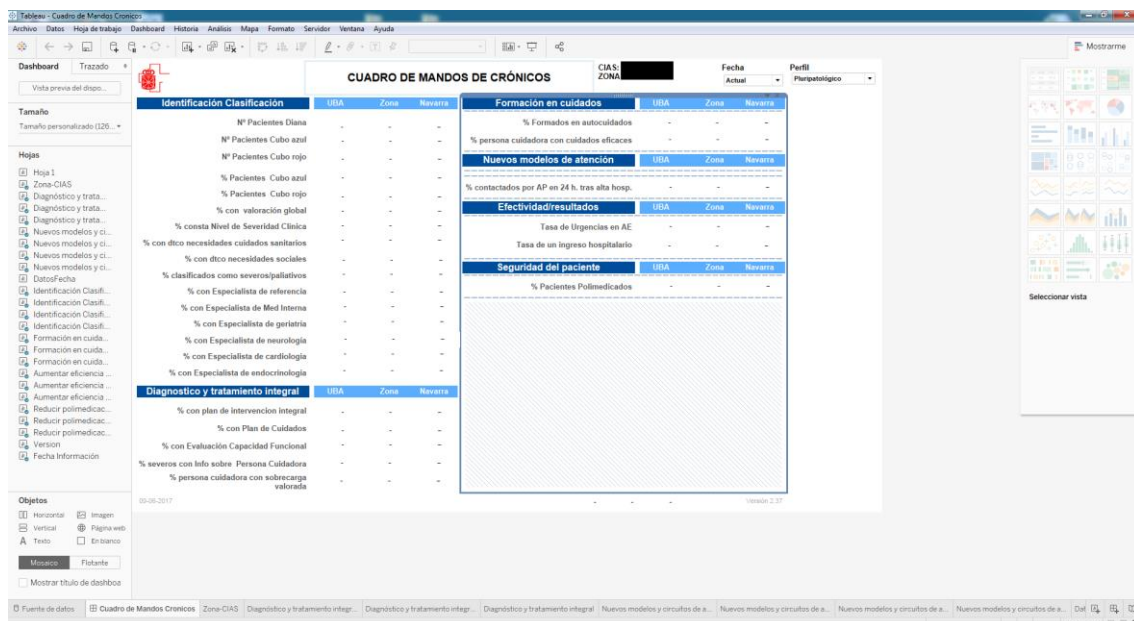


Ilustración 5 Interfaz de la herramienta Tableau

Una vez explicadas las herramientas principales para la generación de informes, a partir de grandes bases de datos, basadas en sistemas de BI, tomamos la elección de la herramienta a emplear a lo largo de este proyecto. En este caso la elección, se realiza mediante un concurso, pues es una herramienta para emplear en Gobierno de Navarra. La ganadora del pliego fue Tableau por ello a lo largo del documento veréis varios cuadros de mando y paneles de control, generados con esta herramienta.

1.5 Propuesta

Para elaborar mi trabajo fin de grado sobre Inteligencia empresarial y sobre cómo utilizar los datos para mejorar el rendimiento de una empresa, me he integrado en un grupo de desarrollo que emplea técnicas de BI para analizar la información de los centros de salud de Navarra y poder explotar mejor los recursos disponibles. Cuando me incorporé al equipo, éste estaba trabajando en la segunda versión, tanto de la base de datos de BI como de los informes. Actualmente estamos trabajando en el desarrollo de una tercera versión mejorada, al mismo tiempo que damos soporte a la versión 2. Más adelante concretaré cómo es cada versión y las mejoras de una a otra.

Para organizarnos seguimos una metodología ágil que nos permite trabajar con las dos versiones simultáneamente sin demasiadas complicaciones. En concreto, seguimos el modelo SCRUM, cada dos semanas realizamos reuniones con el cliente que en adelante denominaré "Negocio" que es un representante de los médicos de los hospitales de Navarra, para presentar el trabajo realizado en las dos últimas semanas. Además de presentar el producto, en esta reunión, Negocio establece nuevos requisitos priorizados que el equipo de desarrollo tendrá que realizar. Tras esta reunión llamada review, se realiza una reunión de planificación en la cual el equipo de desarrollo estima el tiempo que se tarda en realizar cada una de las tareas propuestas por negocio, y escoge según las prioridades y los tiempos las tareas que se llevarán a cabo durante el próximo *sprint*, de dos semanas. En cada *sprint* se intenta dedicar la

mitad de las horas de trabajo del equipo a cada versión, para que ambas versiones convivan y no ahogarnos en el soporte de la versión dos.

Sucede a menudo, que los requerimientos de negocio se centran en la versión 2, esto se debe a que es la que están empleando actualmente los médicos con lo cual donde ven los errores y las nuevas funcionalidades. Es decir, que aparte de ayudarnos a contrastar los datos que mostramos, que esto viene muy bien para el equipo de BI, el hecho de que estén trabajando con una versión antigua no hace más que ralentizar el lanzamiento de la nueva, pues constantemente se solicitan nuevas funcionalidades sobre la versión antigua.

Como ya se ha podido observar, la metodología SCRUM cuya principal ventaja es la posibilidad de adaptarse a requerimientos cambiantes, es la ideal para este desarrollo de BI, pues el cliente cambia constantemente los requisitos. Además de la metodología de trabajo que es muy importante, explicaré que técnicas y herramientas utilizamos para el desarrollo de inteligencia empresarial en el Servicio de Sistemas de Información del Área Sanitaria de Navarra.

2. Conceptos básicos

En este capítulo introduciré algunos conceptos básicos de Business Intelligence. Además de algunos términos propios de la entidad en la que se desarrolla el proyecto.

Sistemas de información

Conjunto de componentes relacionadas que recolectan, procesan, almacenan y distribuyen información utilizada para cubrir una necesidad o un objetivo.

Business Intelligence (BI)

Conjunto de técnicas, metodologías y herramientas que generan información de confianza a partir de datos en bruto. Se utilizan dentro de la organización para ayudar a la toma de decisiones estratégicas y optimización de procesos.

Permite a la entidad recoger datos de fuentes internas y externas prepararlos para el análisis y desarrollar y ejecutar consultas para la creación de informes, cuadros de mando y visualizaciones de datos. Los destinatarios de estos informes son los tomadores de decisiones y trabajadores operacionales de la organización.

Objetivos principales:

- Proporcionar a la empresa información privilegiada que permita identificar al instante los factores que afectan a su rendimiento.
- Crear estrategias más precisas orientadas al futuro.
- Mejorar la eficiencia.
- Aumentar la rentabilidad.
- Mejorar la satisfacción y lealtad de los clientes
- Identificar patrones de comportamiento
- Basados en datos histórico prever tendencias y escenarios futuros

Sistema Operacional (SO)

Se conoce también como sistema OLTP (Online Transactional Processing System). Es un sistema utilizado para procesar el gran volumen de transacciones diarias de una gran organización como lo es Gobierno de Navarra.

Características principales:

- Preservar la integridad de las transacciones y su eficiencia.
- Orientarse a datos actuales con actualizaciones e inserciones permanentes.
- Normalmente asociados a una aplicación, con reducida integración y redundancia de datos
- En la mayoría de los casos no contiene un componente de informes o es bastante reducido.

Sistema de explotación (SE)

Comúnmente identificado como sistema OLAP (Online Analytical Processing System)

Sistema diseñado para poder procesar de forma eficiente el gran volumen de peticiones de análisis de datos que recibe y así atender las distintas necesidades de información de la organización.

Un SE es un producto que aglutina conjuntos de datos heterogéneo para poder dar respuesta a diferentes necesidades de información planteadas por diferentes perfiles de usuarios que lo utilizan.

Características principales:

- Está orientado a datos históricos donde no hay volatilidad (aparte de las cargas periódicas del sistema)
- Es un sistema integrado donde se evita la redundancia de la información.
- Incorpora un sistema de generación de informes donde se presentan los resultados del análisis realizado.
- Sus fuentes de datos habituales son bases de datos OLTP pero pueden recibir información de otros datos estructurados como ficheros formateados o de otros datos no estructurados como pueden ser archivos de texto o documentos sin formato predefinido o emails por ejemplo.

Data Mart (DM)

Repositorio de información integrada, unificada e histórica de una organización.

Es parte del *frontend* del SE y se caracteriza por almacenar un gran volumen de datos.

Objetivos principales:

- Los datos que integra tiene que ser fácilmente entendibles por los usuarios finales.
- Es una solución orientada a la ejecución eficiente de consultas que explotan un gran volumen de información.
- Almacena datos estructurados en tablas de dimensiones y hechos siguiendo el modelo en forma de estrella.
- Es el repositorio final donde se cargan los datos procedentes de los sistemas fuente después de transformados/tratados.
- Es un repositorio disponible para ser consultado por diferentes herramientas e informes (predefinidos en los diferentes Sistemas de Explotación). Cuando el volumen de información existente lo demanda, incorpora tablas agregadas que no son más que rollups de las tablas de hechos y que permiten mejoras significativas de rendimiento.

Esquema en estrella

El Data Mart presenta físicamente los datos en un esquema en estrella.

Dicho esquema contiene una tabla de hechos que se relaciona con una o más tablas de dimensiones en una relación uno para muchos (1-N), tal como se puede ver en el ejemplo presentado en la Ilustración 6.

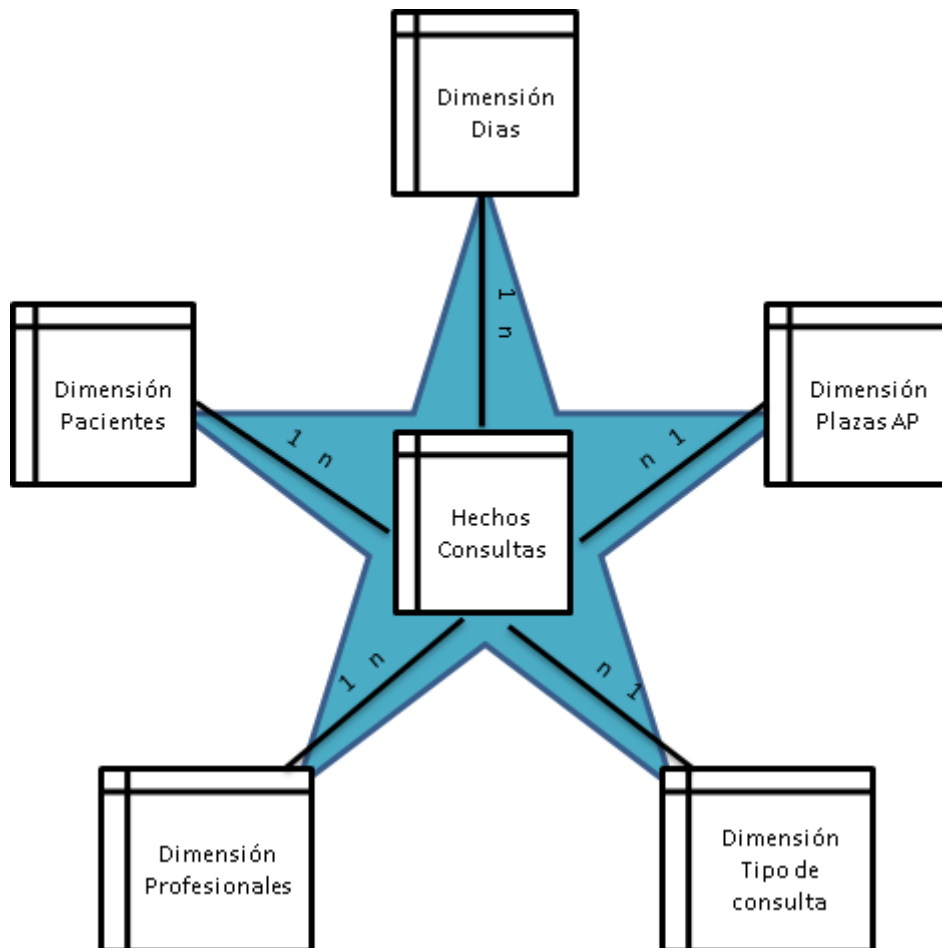


Ilustración 6 Esquema en estrella.

Staging Area (SA)

Area de almacenamiento intermedia que sirve de apoyo a la transformación de datos.

Es parte del backend del SE.

Principales características:

- Zona donde se cargan los datos extraídos de los sistemas operacionales para su posterior transformación y carga en el modelo de datos del SE
- Es normalmente un repositorio temporal por lo que su contenido puede ser eliminado sin que esto afecte la funcionalidad y el rendimiento del SE.

En nuestro caso las áreas de Staging no se eliminan terminada las cargas de los SE pues facilitan futuras validaciones o tareas de *troubleshooting*.

TroubleShooting

El *troubleshooting* es una forma de solucionar problemas, basada en la búsqueda de la fuente del problema, información del error, falta de funcionalidad, etc. Una vez sabemos cómo se reproduce el problema y conocemos el error, eliminamos los componentes innecesarios y verificamos si el problema persiste. Si el problema todavía no se ha solucionado, se pasará a la forma habitual de trabajar con errores y se buscarán las causas comunes del error concreto que afrontamos.

Proceso de transformación de datos

Comúnmente identificado como proceso ETL (Extract, Transform y Loading)

Componente de un Sistema de Explotación que transforma los datos fuente del sistema y los carga en un repositorio de datos.

Características principales:

- Valida los datos fuente contra tablas maestras propias (con datos integrados de diferentes sistemas de información).
- Transforma los datos de entrada realizando tareas de: reformato, limpieza, filtrado, etc...
- Realiza la integración de la información que proviene de diferentes fuentes.
- Carga de la información según un modelo de estructuración de datos predeterminado.

Los procesos de transformación de datos más habituales son ETL y ELT. En la organización esta normalizado el uso de ETL y sus motivaciones se presentan más adelante en el documento.

Réplicas de explotación

Base de datos de lectura que se corresponde a una copia total o parcial de la base de datos del sistema operacional

Sirve de fuente de datos a los Sistemas de Explotación y como base de datos de explotación para el propio sistema operacional.

Objetivos principales:

- Se utiliza para mitigar el impacto en el rendimiento por acceder directamente y recurrentemente a las bases de datos OLTP para realizar tareas consultas de datos pesadas.
- La replicación entre los repositorios OLTP y Explotación puede contener los siguientes tipos de filtros:
 - Filtros a nivel de objetos de bases de datos. Un objeto se puede copiar o no.
 - Filtros a nivel de datos de un determinado objeto. Se puede copiar un rango específico de datos de un determinado objeto.

Repositorio Integrado Normalizado (RIN)

Area de almacenamiento persistente intermedia que sirve de base y repositorio único para la carga posterior de los Data Mart's.

Principales características:

- Es opcional y se recomienda su uso únicamente cuando se dan un conjunto de características descritas en el siguiente apartado de este documento.
- Es un repositorio de se ubica entre las fuentes de datos y los Data Mart's
- Cuando existe es la fuente de datos exclusiva para la carga de los diferentes Data Mart's de la organización
- Es un repositorio persistente que contiene información única por lo que conlleva una gestión específica.
Aquí se guardan tablas y datos de configuración del sistema de explotación, como por ejemplo:
 - mapeo de maestros entre diferentes sistemas operacionales
 - claves primarias de las dimensiones troncales

Cuadro de Mando

Un cuadro de mando es un informe visual que permite tener una visión global de la organización de un vistazo. También existen cuadros de mando de información por áreas, que se generan realizando agregaciones sobre el DM.

Proceso ETL (Extract, Transform and Load)

Es un proceso que se emplea para la extracción de los datos de las bases de datos de origen, el tratamiento de estos datos y la carga en las tablas destino del DM.

3. Análisis

En este capítulo, analizaré la situación anterior a la realización del proyecto del Servicio Navarro de Salud. A continuación, estudiaré los requisitos funcionales y no funcionales del producto necesario a desarrollar en este proyecto de BI. El producto se denomina Sistema de Explotación Poblacional y está compuesto por un modelo de bases de datos que almacena los datos extraídos de las fuentes y un sistema de vistas, que contienen numerosos indicadores.

3.1 Situación actual (antes de la realización del proyecto)

En primer lugar, para poder analizar los requisitos de negocio, necesitamos conocer la situación en la que estaba la explotación de información en el Servicio Navarro de Salud.

La estructura organizativa, implicada en la explotación de la información en el área de Salud es complicada. Pues no existe un punto de referencia único de explotación de la información a nivel general en el SNS-o, sino servicios responsables de ello por áreas. Esta descentralización conlleva la realización de esfuerzos duplicados, para lograr objetivos parciales. Pues es muy complicado que un área extraiga información relevante a nivel general de salud sin datos de otros áreas.

Dada esta necesidad de obtener información de otras áreas, explicaré la gestión de la demanda de información, personas autorizadas, procedimientos establecidos para la solicitud de datos. No existen protocolos establecidos, salvo algunos en casos muy puntuales de modo que esta solicitud se realiza de manera informal por correo electrónico y se resuelven en función de las capacidades del servicio responsable o de la urgencia. De modo que tampoco hay definidos unos tiempos de respuesta.

Pero esta información de la que estamos constantemente hablando, ¿de dónde se obtiene, cuál es su origen? Pues bien, existen multitud de sistemas de origen de la información. Estos sistemas están muy dispersos, algunos casos son SNS-O, Atención Primaria, CHN (Complejo Hospitalario Navarro), Salud Mental, Administración y Servicios Generales... Lo cual conlleva tener un conocimiento claro de que información se encuentra en cada sistema de origen. Además destaca la dificultad de explotar la información en HCI (consultas de más de 6 meses se bloquean). Para extraer cierto nivel de detalle, hay que extraer la información de varias fuentes y posteriormente cruzarla manualmente. Además por lo general, las diversas aplicaciones no están interconectadas, de modo que como he dicho hay que realizar un trabajo manual importante para poder cruzar los datos, sin obtener resultados muy fiables.

En resumen, para obtener la información para realizar una determinada consulta es necesario realizar varias peticiones a diferentes productos, que nos las contesten (dado que no existen plazos ni protocolos esto puede prolongarse indefinidamente en el tiempo), a continuación requiere habilidad para cruzar los datos, cruces no demasiado consistentes. De esta forma podríamos obtener algo de información, como podemos deducir es un trabajo muy costoso y elaborado, que además pueden estar realizando paralelamente en varias áreas, lo cual supone esfuerzos innecesarios.

Una vez se obtienen los datos, como he indicado anteriormente, se pasa a la fase de producir informes. Los informes que se generaban eran normalmente poco visuales y difíciles de comprender. Se generaban cuadros de mando, donde se calculaban los indicadores como he explicado recientemente y posteriormente se rellenaban los ficheros a mano. Si en varias áreas repiten algunos indicadores, tendremos información duplicada en varios informes. Por último la distribución de estos informes, se hacía mediante correos electrónicos y se enviaban informes en pdf, lo cual dificulta el tratamiento de los datos por parte de los destinatarios y es difícil comprobar las leyes de protección de datos, utilizando este medio de divulgación.

En Gobierno de Navarra, y más insistentemente en el ámbito de Salud es muy importante el cumplimiento de la LOPD. De modo que es fundamental llevar un control de los datos, de quien puede ver que informes, que datos muestra cada informe, si hay diferentes informes con el mismo indicador y los resultados que muestran son diferentes...

A partir de esta introducción de la situación inicial, generamos los objetivos principales.

3.2 Análisis DAFO

A partir de las conclusiones extraídas en el apartado anterior, se ha construido una matriz DAFO, en la ilustración 7, que analiza la situación actual de la explotación de la información en Salud.

En este análisis DAFO, en concreto, las fortalezas son los elementos facilitadores y palancas para avanzar de los que ya se dispone; las oportunidades son aquellos factores positivos que surgen en el entorno y que, una vez identificados, pueden ser aprovechados; las debilidades son los elementos a corregir o mejorar; y las amenazas, el conjunto de aspectos a mitigar o reducir.

FACTORES A MEJORAR	FACTORES A POTENCIAR
DEBILIDADES	FORTALEZAS
<p>Descentralización de la explotación de la información que conlleva la realización de esfuerzos e iniciativas independientes en cada área</p> <p>Proceso manual y poco estandarizado que deja escaso margen de tiempo para el análisis de datos</p> <p>Información dispersa en varios sistemas de origen</p> <p>Definición no unívoca de determinados indicadores en los sistemas de origen</p> <p>Dependencia de terceros para recibir cierta información requerida</p> <p>Dificultad para alcanzar cierto nivel de desagregación de la información</p> <p>Información orientada a la actividad, no a resultados</p> <p>Falta de transversalidad de los datos (actualmente se refieren a cada ámbito)</p> <p>Presentación de datos poco visual e intuitiva</p> <p>Escaso feedback de los usuarios finales</p> <p>Distribución de los CdM/informes que depende en gran medida del envío de emails y que, en algunos casos, no se realiza al personal clínico</p> <p>Falta de CdM clínicos</p> <p>Herramientas de análisis no implantadas en todas las áreas clave</p> <p>Falta de enfoque a la mejora continua</p> <p>Inexistencia de un sistema de alarmas que avise de desviaciones en los objetivos marcados</p>	<p>Identificación clara de peticionarios de información y del tipo de información que solicitan</p> <p>Existencia de conocimiento funcional y técnico previo sobre explotación de la información</p> <p>Identificación realista de debilidades y aspectos de mejora del actual proceso</p> <p>Implicación de los profesionales en el proceso de explotación de la información</p> <p>Existencia de cultura de evaluación de resultados y calidad de los servicios en Atención Primaria</p>

Dificultad para identificar tendencias, patrones, etc. en el ámbito sanitario	Establecimiento de canales formales y centralizados para realizar peticiones de información
Riesgo de no atender a tiempo y de forma adecuada las peticiones de información	Priorización de las demandas de BI
Agilidad limitada en la toma de decisiones	Almacenamiento único de datos
Dificultad para disponer de información que apoye nuevas actuaciones e iniciativas	Definición única y común de indicadores
Riesgo de basar decisiones en datos no fiables	Calidad de los datos que permite tomar decisiones basadas en información veraz
	Datos obtenidos de manera sencilla y ágil
	Posibilidad de realizar análisis transversales y predictivos
	Información orientada a los clínicos
	Contenido, formato, etc. de los CdM/informes definidos por el responsable de negocio y adaptados a sus necesidades
	Detección automática de desviaciones o irregularidades
AMENAZAS	OPORTUNIDADES

Ilustración 7 Tabla Análisis DAFO

3.3 Objetivos principales de negocio

A continuación muestro las necesidades que vamos a analizar, de las que habrá que calcular el esfuerzo necesario para implementarlas y se establecerán los trabajos que se desarrollarán para realizar dicha implementación. Se ha explicado anteriormente el estado de cada una de las necesidades de negocio y a continuación se planteará como afrontar cada una de ellas.

Estas son las necesidades principales de GB, en cuanto a la explotación de la información:

- Modelo único de información
- Modelo de Gobierno del dato
- Gestión centralizada de la demanda
- Comunicación y divulgación
- Capacidad de análisis rápido, ágil y sencillo
- Transversalidad
- Gestión de la seguridad
- Metodología unificada
- Creación de las bases de datos que integren los productos principales de GB
- Creación de vistas y cálculo de indicadores
- Creación de un informe para médicos

- Distribuir el informe para médicos
- Aplicar la seguridad en la distribución del informe para médicos
- Rendimiento óptimo del ETL

3.4 Resultados análisis

Una vez analizada la situación inicial y los objetivos que se quieren lograr, se decide crear un modelo de datos unificado, empleando modelos de BI, e informes más elaborados y visuales, empleando herramientas especiales para ello. Para realizar este proceso, se plantean dos situaciones iniciales, una es contratar a una empresa para que realice todo el desarrollo y la otra crear una unidad central de BI en Gobierno de Navarra. Ambas opciones se estudiaron en profundidad y se llegó a la conclusión de generar un equipo propio de BI para Salud. La principal ventaja de tener un grupo interno, y no contratar el modelo completo, es que acepta cambios y podemos obtener resultados incrementalmente. Es decir, contratar a una empresa que realice el modelo total, conllevaría varios años de trabajo en los cuales no podríamos contemplar ni un resultado, mientras que formando un equipo propio podemos obtener resultados parciales, mediante el empleo de la arquitectura de Kimball y la generación de Data Mart's según necesidades.

En definitiva, dado que se ha optado por la creación de una unidad central de BI para Salud, está asumiendo las siguientes funciones.

- Impulsar el desarrollo de Sistemas de Evaluación e Inteligencia Sanitaria
- Evaluar el Plan de Salud y el Plan Estratégico del Departamento de Salud
- Garantizar la coherencia técnica y la adecuada orientación estratégica de todos los Cuadros de Mando de la organización
- Garantizar la adecuada gestión del dato
- Asegurar la adecuada definición, construcción y utilización de los indicadores estratégicos clave en toda la organización
- Optimizar las herramientas y tecnologías evitando duplicidades

Además, el diseño y desarrollo de esta unidad ofrecerá al Departamento de Salud los siguientes beneficios:

- Alineamiento con la estrategia global del Departamento de Salud.
- Simplificación del modelo operativo, organizativo y de gobierno al aumentar el nivel de automatización y monitorización.
- Mayor coordinación entre las diferentes iniciativas y áreas, evitando esfuerzos duplicados.
- Optimización, normalización e innovación en los procesos de explotación de la información.
- Facilidad de relacionar diferentes datos procedentes de diferentes sistemas operacionales.
- Aumento de la calidad y garantía de la información manejada.
- Mayor agilidad y flexibilidad en la satisfacción de la demanda, reduciendo los tiempos de respuesta.

- Mayor integración y reutilización de la información y aplicativos, evitando incongruencias en los resultados.

En el próximo capítulo, se describirán los principales procesos que se deben realizar en la nueva unidad de BI para salud, para dar respuesta a todos los servicios y tareas necesarias para llevar a cabo el proceso de explotación de la información de la manera más eficiente posible.

4. Diseño

A lo largo del siguiente apartado se explica cómo se ha diseñado la arquitectura del sistema de BI, basándose en los análisis previos. Para la arquitectura se han estudiado dos modelos tradicionales, el modelo dimensional de Kimball y el modelo normalizado de Inmon.

4.1 Modelo normalizado vs modelo dimensional

Una vez ya presentado el proyecto, llamado evaluación de resultados, diseñaremos la arquitectura necesaria para llevarlo a cabo. Para analizar qué arquitectura se adapta mejor a los requerimientos del sistema de explotación estudiaremos el modelo normalizado de Inmon [7] y el modelo dimensional de Kimball [6]. Este modelo dimensional se basa en el Ciclo de Vida Dimensional del Negocio, también definido por Kimball. Este ciclo de vida consta de cuatro principios básicos:

- Centrarse en el negocio.
- Construir una infraestructura de información adecuada.
- Realizar entregas en incrementos significativos.
- Ofrecer la solución completa.

El primer principio, centrarse en el negocio, se refiere a que lo más importante y lo que debemos hacer al construir la arquitectura es focalizarnos en los requisitos de negocio. Para ello realizar un esfuerzo importante en construir relaciones sólidas con el negocio, lo cual nos ayudará a comprender mejor los requisitos que nos presenten.

El segundo principio de Kimball, se refiere a diseñar una base de información única, integrada, fácil de usar, de alto rendimiento donde se reflejará la amplia gama de requerimientos de negocio identificados en la empresa.

El tercer principio referente a la realización de entregas en incrementos significativos consiste en crear el almacén de datos por partes, pero estas partes tienen que ser coherentes y útiles para el cliente. Esto puede recordarnos a la filosofía de las metodologías ágiles y es que van de la mano, salvo que los plazos de construcción y entrega en este caso son más amplios, de entre 6 y 12 meses.

En cuanto al cuarto principio, cuando decimos que se proporciona una solución completa nos referimos a que no solo se entrega una base de datos, sino también los elementos necesarios para que el usuario la utilice. Esto incluye herramientas de consulta ad hoc, aplicaciones para informes y análisis, soporte... Estas herramientas varían mucho de un proyecto a otro, pues dependen de las necesidades de cada proyecto.

Esta metodología se emplea para la construcción de un almacén o fuente de datos integrada, no volátil y variable en el tiempo que ayuda con la toma de decisiones en la entidad que así lo requiera. Este almacén de datos también se suele denominar *data warehouse* o DW.

El ciclo de vida comienza con la planificación del proyecto, en el cual se define el alcance, se identifican y programan las tareas, se planifica el uso de recursos, se asignan las tareas a estos

recursos y se elabora un plan de proyecto que contenga todo lo planificado. Además, se lleva una monitorización del estado de los procesos y en caso de que no se vayan cumpliendo los objetivos o cortos alcances se realiza un rastreo de problemas.

Tras la planificación a gran escala del proyecto, se definen los requerimientos de Negocio. Para este proyecto, como ya he indicado anteriormente, los requerimientos y la planificación detallada se van realizando *sprint* a *sprint* siguiendo la metodología *Scrum* cada dos semanas. Pero todos estos *sprints* están dentro del plan general mencionado anteriormente donde se indica el alcance del proyecto, en cuanto al tiempo y a los requerimientos se refiere.

En definitiva, en nuestro caso trabajamos con la metodología *Scrum* para organizar el trabajo del equipo día a día, pero estas tareas que realizamos en cada *sprint* se obtienen de dos fuentes, una sería el plan general que hemos desarrollado al inicio del proyecto según indica el modelo de Kimball y otra fuente son las peticiones que negocio realiza en cada *sprint*.

Una vez se tienen unos requerimientos de negocio y un plan sobre el que trabajar, pasamos al modelado dimensional. Elegimos el proceso de negocio, que depende de la dirección y del plan. Establecemos la granularidad, debemos enfocar el nivel de detalle del DW, esta granularidad dependerá de los requisitos de negocio. En nuestro caso, como estos requisitos son cambiantes, conviene diseñar el DW al mayor nivel posible para luego realizar los agrupamientos que se soliciten. Elegimos las dimensiones, normalmente al inicio el equipo debate sobre las dimensiones necesarias para cumplir con el nivel de granularidad anteriormente indicado y según la matriz de procesos/dimensiones. Normalmente al inicio se estiman unas dimensiones, que se suelen modificar a medida que se desarrolla el proyecto. A continuación se identifican las medidas y las tablas de hechos. Cuando hablamos de medidas hablamos de los atributos de las tablas de hechos que serán empleados por los usuarios. Podemos observar en la ilustración 8 un esquema de la arquitectura de Kimball, que ayuda a la comprensión de esta arquitectura.

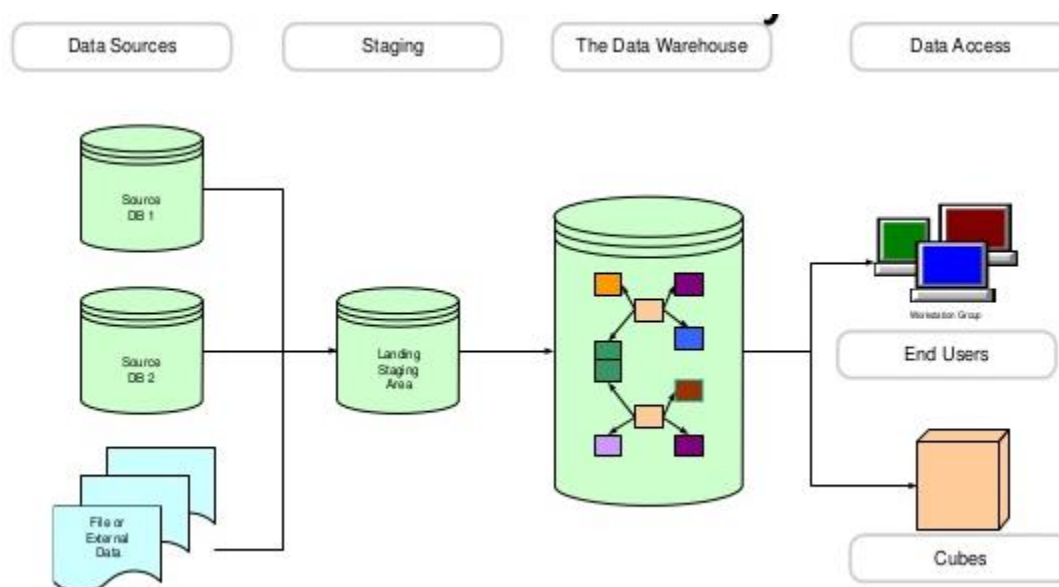


Ilustración 8 Arquitectura de Kimball

La arquitectura de Inmon trata de conseguir un Corporate Information Factor (CIF), es decir, un almacén de datos único y global de toda la empresa. Este CIF es orientado a temas,

contiene datos integrados, no es volátil y es variante en el tiempo. Este método realiza en primer lugar la extracción masiva de los datos de todas las fuentes, los pasa por ETL y carga en un *Staging Area* (SA). Dentro del SA se validan, y por último se pasan al *Data Warehouse*, donde se consolidan los datos y se añaden metadatos.

Para Inmon lo más importante es que el modelo de datos esté construido en tercera forma normal gracias a lo cual se obtienen datos no redundantes, con integridad referencial, de fácil mantenimiento, además de ayudar a reducir el tamaño de la base de datos. Pero todo ello supone un coste, y es que las consultas para analizar el DW pueden resultar muy complejas. Por este motivo, a partir del DW con el conjunto global de datos de la empresa normalizados, se obtienen los diferentes Data Marts (DM) que facilitan las consultas. Este sistema también es conocido como Top-Down porque empieza desde arriba, que sería el DW, para luego bajar a cada DM temático. El ejemplo ideal para comprender los DW y los DM dentro de una empresa sería que un DW contiene toda la información (la que consultan los CEO), y un DM contendría la información de un departamento. La mayor ventaja de esta arquitectura es que todos los datos de los diferentes DM están conectados al DW, de manera que los datos de los distintos departamentos se pueden comparar sin incongruencias. Podemos observar en la ilustración 9 un esquema de la arquitectura de Inmon, que ayuda a la comprensión de esta arquitectura

Observando las ilustraciones 8 y 9 podemos apreciar la diferencia entre la estructura de la base de datos del DW, cuya prioridad ha sido seguir las reglas de la 3FN, y la base de datos del DM que es un modelo dimensional con datos agregados y muy útil para el análisis.

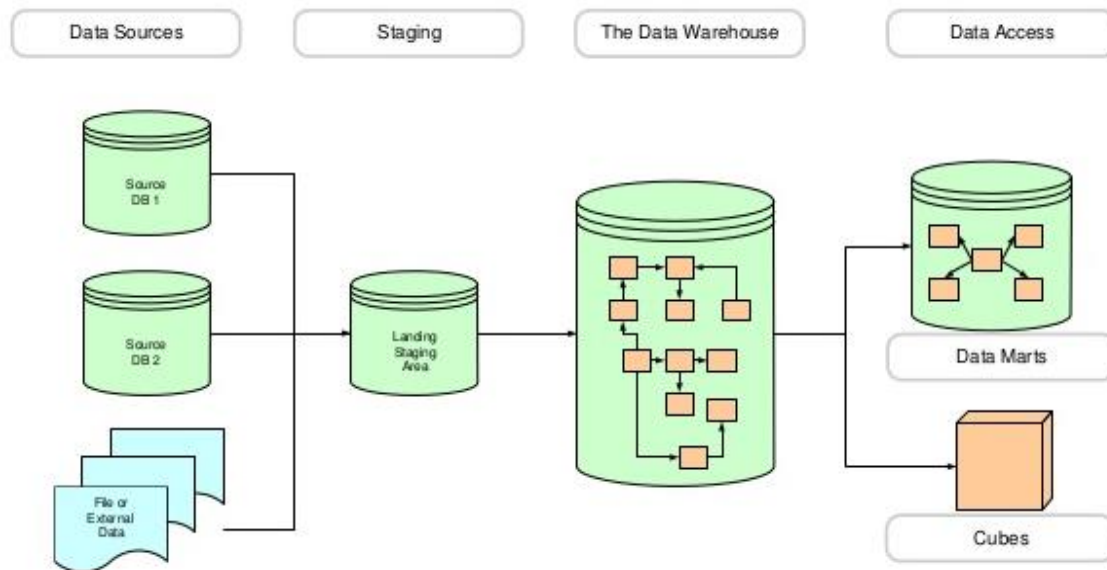


Ilustración 9 Arquitectura de Inmon

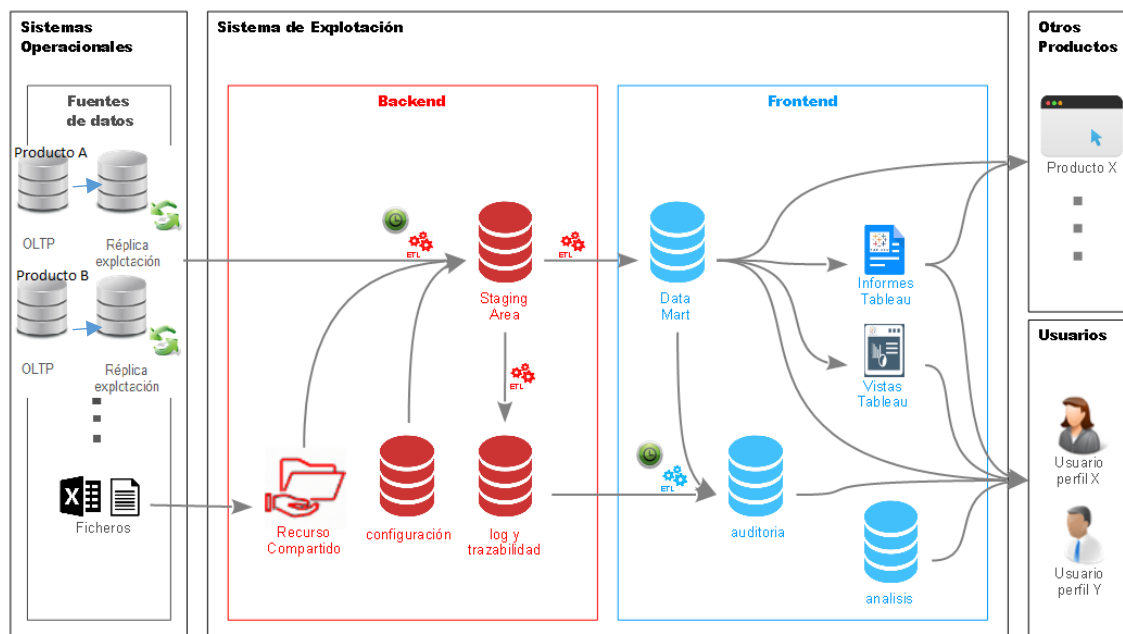
A la hora de diseñar la arquitectura de un sistema de BI, el primer dilema al que algunos bautizan como el “eterno debate de BI” es elegir si construir un modelo dimensional o uno normalizado. Pues bien, depende de cada situación. A grandes rasgos, podríamos decir que Kimball es un método más flexible si no se tiene una idea certera de lo que es un sistema de BI o si se desconocen los datos que se poseen. Otra ventaja de Kimball es que se puede implementar poco a poco, de DM en DM, mientras que utilizando Inmon es necesario montar

todo el DW para luego extraer los DM. En resumen, Inmon es adecuado para sistemas complejos y asegura perdurabilidad y consistencia, mientras que Kimball sería una metodología para proyectos más pequeños o proyectos que primen la usabilidad.

Tras esta breve comparación de las dos metodologías, hemos tomado la decisión de trabajar siguiendo una metodología Kimball debido a condicionantes como que los productos de los cuales extraemos los datos no están documentados y están altamente desnormalizados. El usuario final desconoce el conjunto de la información gestionada, luego no le interesa un modelo completo pues nadie lo va a utilizar, sino pequeños DM más concretos. Además, cuando se encargó el proyecto en 2014, como ya he mencionado en la introducción, se buscaban resultados en un periodo corto de tiempo, de menos de 6 meses. Por último, otro requisito fue que el modelo de datos fuera comprensible para los usuarios analistas, que no son informáticos, luego les conviene un sistema DM sobre el que se realizan consultas más sencillas, que un sistema DW que no sabrían explotar. Por todos estos motivos se tomó la decisión de realizar un desarrollo *Bottom-Up* basado en el modelo dimensional de Kimball. Siguiendo las pautas que éste explica en [6].

4.2 Arquitectura de los Sistemas de Explotación

En la Ilustración 10 se muestran los principales elementos que interaccionan con los Sistemas



de Explotación. A continuación explicaré por partes dichos elementos.

Ilustración 10 Arquitectura de los sistemas de explotación

4.2.1 Sistemas operacionales

Los sistemas operacionales (SE) están compuestos por la base de datos OLTP y ficheros. Las bases de datos OLTP son un repositorio de datos de las aplicaciones corporativas que sirve como fuentes de datos para los sistemas operacionales. Esta fuente puede ser directa o

indirecta. En nuestro caso es indirecta, pues creamos réplicas de explotación de estas bases de datos.

Como ya he mencionado, utilizamos réplicas de explotación en lugar de directamente los datos de las aplicaciones por varios motivos. En primer lugar, el acceso del SE a las distintas bases de datos se realizará de manera puntual y controlada. Este acceso será la creación de las réplicas. El segundo motivo es que el uso por parte del SE sobre las BBDD de los operacionales no impacte en su rendimiento. En tercer lugar, la existencia de las réplicas también favorece a los propios operacionales, pues ellos mismos las pueden utilizar con objetivos de explotación. Los inconvenientes de las réplicas son claros, aumenta la latencia en la que los datos pasan de los sistemas fuentes a nuestro DM y el incremento del volumen de datos pues tendremos todo por duplicado.

Hemos de mencionar, que el equipo que se encarga del desarrollo de BI no es el encargado de generar las réplicas, sino que cada producto es responsable de generar las réplicas de las bases de datos de sus aplicaciones. No todos los productos siguen las mismas normas, hay algunos que realizan copias completas del sistema, mientras que otros solamente copian parte de la estructura. Estas copias han de seguir unos requerimientos mínimos definidos por el SE, como por ejemplo realizarse mensualmente. Otra directriz, es la nomenclatura. Todas las réplicas poseen el sufijo _EXPLOT y se ubican en servidores corporativos dedicados a tareas de explotación.

En cuanto a las otras fuentes, también utilizamos ficheros de datos. El formato de estos ficheros y la ubicación son responsabilidad del producto, pero se encargan de comunicar los cambios al equipo de BI para poder mantenerlos como fuente de datos.

4.2.2 Sistemas de Explotación - Backend

En los próximos apartados se detallan los elementos específicos del backend del Sistema de Explotación.

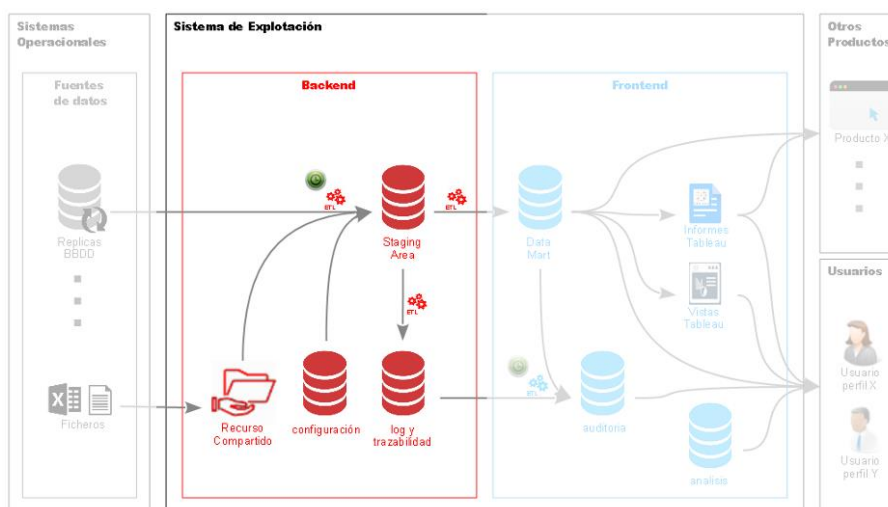


Ilustración 11 Arquitectura de los sistemas de explotación, Backend

Como podemos apreciar en la Ilustración 11, el backend está conformado por varias bases de datos. Comenzaré explicando la BBDD de log y trazabilidad. Es un repositorio que incluye información acerca de los *logs* de ejecución, errores y *warnings*, y rendimiento de los diferentes procesos de ETL del SE. ¿Qué motiva su uso? Muy sencillo, una base de datos con todos los *logs* de los distintos procesos permite recoger en un único punto todos los *logs* de las diferentes cargas. Lo cual ayuda a centralizar las tareas de verificación de los estados de los procesos y obtener estadísticas de los procesos de carga, para posibles mejoras.

En la Ilustración 11 también apreciamos una base de datos para la configuración que recoge los datos de configuración global de los diferentes procesos de ETL del Sistema de Explotación. Los motivos para el uso de un repositorio de configuración son similares a los del repositorio de *logs*; el poder almacenar en un único punto todos los datos de configuración y, sobre todo, el disponer junto con los datos de configuración de documentación actualizada de los procesos de carga del SE.

Por último, el *Staging Area*, que es un repositorio de los datos del SE. Se utiliza para el proceso de transformación. Aquí realizamos las tareas de limpieza, integración y unificación de datos que se cargarán posteriormente en el DM. Esta base de datos es temporal, lo cual permite su borrado al acabar la carga del DM sin afectar al funcionamiento del producto. Esto facilita mucho el trabajo de los desarrolladores. Los motivos por los que en BI se suele emplear una base de datos denominada SA para el procesamiento de los datos son numerosos. A continuación mencionaré algunos de los más importantes en nuestro caso.

- Reducir el impacto en las fuentes de datos, para no ocasionar problemas de rendimiento.
- Mayor flexibilidad en los procesos de limpieza, validación e integración antes de cargar la información en los Data Mart. Además en cuanto a las validaciones permite realizarlas antes, durante y después de la carga.
- Mayor control en las tareas de *troubleshooting*. Ya que gracias al SA, tendremos el proceso de tratamiento de los datos dividido en más bases de datos, lo cual facilita el proceso de búsqueda de errores, acotando el dónde buscar.
- Mejor reacción ante errores, cuando falla un proceso no es necesario recargar los millones de datos con los que trabajamos, sino que es suficiente con volver a lanzar los procesos desde el fallo de forma que todos los datos volcados antes del error se mantienen consistentes. Esto es muy útil debido a que todas las tareas de ETL están perfectamente separadas, de modo que si por ejemplo falla la Transformación, no será necesario repetir el proceso de Extracción realizado.

En definitiva el SA es la mejor forma de realizar las tareas de ETL sin incrementar exponencialmente el tiempo de ejecución y el rendimiento de la solución. Claramente crear una copia de toda la BBDD también tiene inconvenientes, por un lado la necesidad de mayor espacio físico y por otro lado el aumento de la latencia para pasar los datos desde las fuentes de origen hasta el DM.

A lo largo del trabajo he mencionado en numerosas ocasiones el procesamiento de los datos, el ETL, procesos que explicaré en el apartado de bases de datos del modelo de BI, dentro del capítulo de desarrollo.

Por último, tenemos el Repositorio Integrado Normalizado (RIN) que es un repositorio que permite utilizar un único sistema de ETL para la carga de diferentes DM. También permite una gestión integrada de tablas maestras, mapeando los valores entre distintos sistemas operacionales. Además de la creación de claves para todas las dimensiones, iguales en los distintos DM y persistentes, es decir, que no cambian con el tiempo. Además, permite la gestión de las *historificaciones* (historificar quiere decir que cuando lleguen datos actuales, los almacenaremos pero sin borrar los que había antes en la base de datos si no que ambos datos convivan en la base de datos con un indicador que nos diga cuál es el más actual. En salud tenemos claros ejemplos de la necesidad de historificar por ejemplo para obtener el historial clínico de un paciente o para poder obtener la evolución de una enfermedad en un paciente.) Independiente de los sistemas operacionales. Por último, permite gestionar cargas completas, incrementales o parciales de una misma tabla de hechos. Ya he mencionado muchas de las ventajas que nos aporta este repositorio y podemos observar que es ideal cuando tenemos gran cantidad de fuentes de datos, como es nuestro caso. Es ventajoso el empleo del RIN si tenemos que realizar análisis e integración de datos transversales, es decir, que se obtienen en distintas aplicaciones. En nuestro caso esto sucede a menudo, por ejemplo en la obtención de la dimensión de pacientes participan numerosas fuentes, si queremos saber los datos de urgencias y los datos de un ingreso en AP (Atención Primaria) de un paciente, tendremos que conseguir unir los datos del mismo paciente en las distintas fuentes. Para estos tratamientos de “unificación” conviene trabajar con el RIN.

4.2.3 Sistema de Explotación – *Frontend*

En este apartado se explica la arquitectura del *Frontend* del sistema de explotación.

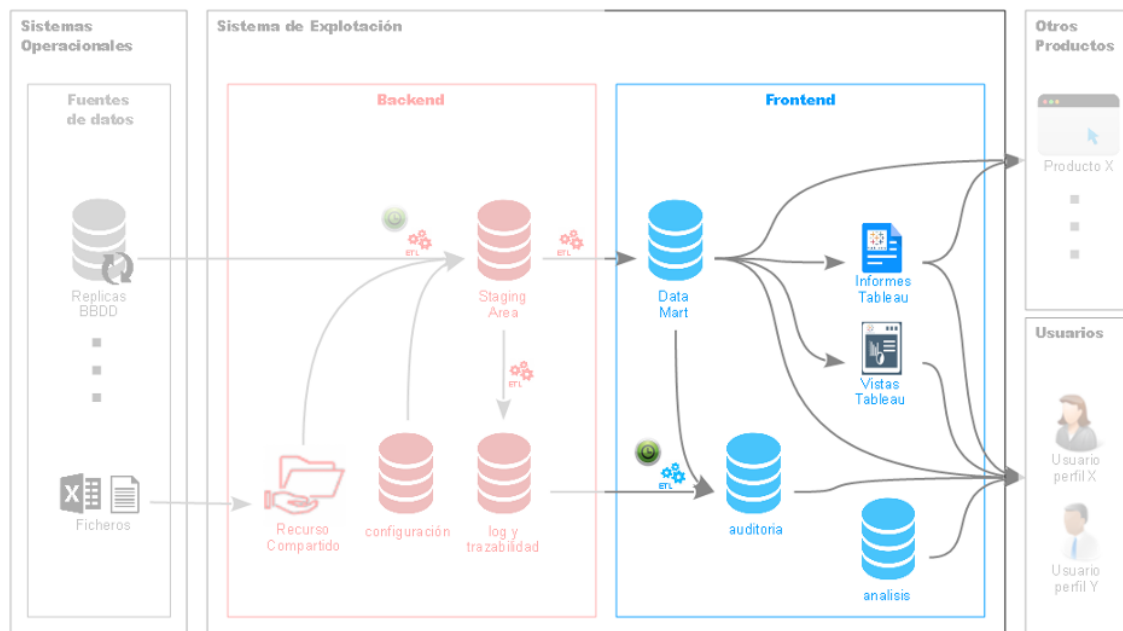


Ilustración 12 Arquitectura de los sistemas de explotación, Frontend

Como podemos ver en la Ilustración 12, el *frontend* está compuesto por varias bases de datos. La más importante es la BBDD Data Mart, pues es la que da servicio a los usuarios finales. Sigue las prácticas definidas por Kimball y el modelado dimensional. Tal y como indica Kimball, la base de datos de DM sigue un esquema en estrella.

En el *frontend* también disponemos de una base de datos de auditoría, este repositorio incluye información acerca de las incidencias en los datos que tratamos en los diferentes procesos del ETL. Se trata de hacer un sistema que se “retroalimente o autocorrija”, esto quiere decir que los datos que se encuentren que no sean correctos, se traspasen a su fuente de origen para que estas puedan corregir el fallo. También se derivan los datos en blanco, para que puedan ser completados por las propias aplicaciones en un futuro. Evidentemente, no todos los datos erróneos o vacíos se derivan a auditoría, pues el tamaño de esta base de datos crecería demasiado como para poder gestionarla, sino que se realiza un filtrado y únicamente se cargan en auditoría los datos erróneos importantes, como podría ser el DNI o el CIPNA (Código de Identificación Personal Navarra, se corresponde con el número de la tarjeta sanitaria) de un paciente. Adicionalmente a este filtrado, se almacena la jerarquía del tipo de incidencia detectada. La principal ventaja de esta base de datos es el disponer de todas las incidencias de datos de los diferentes procesos de carga del SE en un único punto. Además de las incidencias, en estas tablas también almacenamos información sobre los datos.

Así mismo, en algunas circunstancias concretas interesa una base de datos agregada, en la Ilustración 8 aparece bajo el nombre de “análisis”. Esta base de datos es una agregación de la información del DM, previa a los cálculos que se realizan sobre los datos. Esta base de datos se empleará cuando realicemos recurrentemente en periodos cortos de tiempo una o varias consultas con mal rendimiento. Existen varias soluciones como las tablas de agregados o cubos OLAP. Nosotros nos decantamos por las tablas de agregados, que permiten tener físicamente información resumida de los hechos, además del que era su objetivo, mejorar el rendimiento de las consultas con un nivel de granularidad similar o superior al escogido para crear las tablas. Estas tablas de agregaciones se realizan periódicamente a la vez que se refresca el DM para que no queden desactualizadas. Por otra parte, los cubos OLAP tienen un volumen de trabajo mucho mejor, son más fáciles de desarrollar, mantener y realizar *troubleshooting*. La mayor ventaja de esta solución es que se implementa en el mismo almacén de datos con la misma tecnología, luego nunca hay pérdida de servicio parcial como podría ocurrir con una solución de cubos OLAP.

Actualmente en el proyecto no se emplean cubos OLAP pero tampoco se descarta emplearlos en un futuro. Hasta el momento no han sido necesarios, ni nos han aportado ventajas. Un cubo OLAP, lo que hace es como un pre-procesado de los datos agregados. Pre-calcula los datos para todas las agregaciones posibles, si por ejemplo tenemos un modelo con 5 dimensiones, donde cada una tiene dos jerarquías, el cubo OLAP calculará los datos para las 10 agregaciones posibles. Este cálculo previo a la solicitud del dato, como podréis imaginar, hace que cuando necesitemos un dato este se obtenga al instante, pues como ya he dicho ya está calculado. Esta es la mayor ventaja de los cubos, pero también supone una gran desventaja y es que tiene un alto coste temporal todo el cálculo de los cubos. En nuestro sistema los cubos supondrían aumentar el coste temporal en aproximadamente un 30%, esto añadido al coste temporal que tienen los ETL, sería muy alto, por ello no se emplean cubos OLAP. Además, porque tampoco existe un requerimiento por parte del cliente, por aumentar la velocidad de las consultas. (Ni quejas de que estas sean muy lentas.) De hecho, los cubos OLAP poco a poco

se están empleando menos, debido al coste mencionado y al avance de herramientas como Tableau, que permiten cargar en memoria los datos y mostrarlos sin demorarse demasiado. Esto antes era impensable y por eso se empleaban tanto los cubos OLAP.

5. Metodología empleada

En Gobierno de Navarra se trabaja con distintas metodologías según el equipo de trabajo y según los proyectos. En este caso, el equipo de Business Intelligence emplea la metodología ágil Scrum. Esta metodología no se sigue al cien por cien sino que la hemos adaptado a nuestras necesidades, a continuación explicaré la adaptación que hemos realizado. El anterior equipo que desarrollaba tecnología BI trabajaba con una metodología tradicional con la que el cliente estaba muy insatisfecho, por este motivo se cambió de metodología y ahora el cliente está mucho más implicado en el proyecto.

Además, he de mencionar que gracias al buen funcionamiento de esta metodología en nuestro equipo, se está implantando en otros, los cuales vienen como oyentes a algunas de las reuniones de Scrum que realizamos para aprender y emplear también la metodología ágil.

El equipo está formado por el responsable del programa, el responsable técnico (RTec), el gestor de producto (GProd) y el equipo de desarrollo. En las siguientes líneas cuando mencione la palabra jefe, me referiré al responsable técnico.

5.1 Scrum

El equipo de BI sigue una metodología ágil, en particular la conocida como Scrum. Esta metodología se suele emplear para la gestión de proyectos cambiantes, aquellos cuyos requisitos no se conocen al inicio o que se conocen pero hay un alto riesgo de que varíen. De hecho para este tipo de proyectos sometidos a cambios se ha demostrado que una metodología tradicional, en la que se invierte inicialmente mucho tiempo en la planificación de todo el proyecto y posteriormente se trabaja sobre esta estricta planificación, no es útil. De hecho en numerosas ocasiones precipitan el proyecto al fracaso, debido a las dificultades que se encuentran para adaptarse a nuevas funcionalidades que pueda solicitar el cliente, de forma que su plan inicial se tendría que ver modificado, con lo cual todo el tiempo dedicado al inicio acabaría siendo tiempo perdido.

Nuestra adaptación de Scrum, consiste en realizar periodos de trabajo de dos semanas aproximadamente. Al inicio de cada periodo, o *sprint*, realizamos una reunión con el cliente, para definir los requisitos. A continuación realizamos una reunión del equipo de desarrolladores junto con el jefe de proyecto para realizar la planificación de las tareas necesarias para realizar las solicitudes de negocio (cliente). Una vez realizada la planificación la llevamos a cabo, trabajando durante dos semanas. A lo largo de cada jornada de trabajo, se realiza una breve reunión conocida como *daily* en la que cada miembro del equipo informa sobre el estado de la tarea que está realizando. Transcurrida la primera semana de trabajo, el gestor de proyecto comunica al cliente el estado del *sprint*, si intuimos que vamos a alcanzar todos los objetivos o en caso de que no, que tareas se cree que quedarán fuera, por si el cliente quiere “repriorizar” estas tareas.

Por último, al finalizar las dos semanas de trabajo se realiza una reunión con el cliente, en la que se le muestra las funcionalidades que había pedido completadas. Estas reuniones son muy interesantes para conocer mejor al cliente, y que el cliente conozca al equipo y el producto que

este está desarrollando. De manera que en cualquier momento, si por un fallo de comunicación o por unas especificaciones demasiado abiertas, el producto desarrollado no es a lo que el cliente deseaba, se puede cambiar sin perder demasiado tiempo. A medida que el equipo ha ido utilizando esta metodología y estas reuniones, esto sucede cada vez menos pues conocemos al cliente y lo que este quiere. Además nuestro cliente está totalmente abierto a que si durante el desarrollo surgen dudas sobre las especificaciones contactemos con él. Lo cual facilita mucho el trabajo.

5.2 Reuniones

En este apartado explico detalladamente todas las reuniones mencionadas en el apartado anterior, donde se explicaba cómo hemos adaptado la metodología ágil a nuestro entorno.

5.2.1 Reunión con el cliente

Es una reunión en la que el jefe y el gestor del equipo se reúnen con los clientes para determinar las historias de usuario que se quieren abordar en la siguiente quincena. En esta reunión es muy importante asegurarse de entender los requerimientos del cliente para luego explicárselos a los desarrolladores. Además de explicar extensamente las nuevas funcionalidades que el cliente necesita, a cada historia de usuario se le asigna una prioridad, que luego será fundamental para la planificación de las tareas a desarrollar durante el siguiente *sprint*.

5.2.2 Reunión de planificación

En esta reunión participan los desarrolladores y el jefe del proyecto y se desarrolla de la siguiente manera. En primer lugar el jefe de proyecto nos explica de forma general las nuevas historias de usuario que le ha transmitido el cliente.

A continuación, cada historia de usuario se descompone en tareas y estimamos entre todo el equipo cuanto nos costará realizar cada tarea. Para cada estimación de cada tarea, el primero en estimar es el jefe, a continuación cada desarrollador, dejando en último lugar a los desarrolladores que hayan realizado tareas similares. Una vez tenemos todas las estimaciones sobre la mesa, si son muy dispares, quienes hayan dado la menor y la mayor argumentan su estimación y en función de estas explicaciones cada miembro del equipo se replantea su estimación. Acto seguido es el jefe quien escoge la estimación, basándose en lo que ha dicho el equipo. No siempre sigue una regla, hay veces en las que realiza la media de todas las estimaciones, otras veces (si por ejemplo se trata de una tarea con la que no estamos habituados) escoge la mayor estimación y otras veces realiza una media ponderada donde da más valor a la estimación del desarrollador que ha realizado con anterioridad tareas similares. Una vez se han estimado todas las nuevas tareas solicitadas por el cliente, se escogen las que se van a realizar en este periodo.

Para ello se calculan las horas totales que va a trabajar el equipo y por prioridades se van escogiendo las tareas a realizar. Muchas veces ocurre que las dos tareas principales abarcan un gran número de horas de modo que no es posible completarlas dentro del mismo *sprint*, luego se elegirá la más prioritaria y a continuación las siguientes prioritarias cuya duración no supere las horas del *sprint*. En verdad para realizar este proceso no se sigue una métrica estricta, si no que más bien la intuición sobre que combinación de tareas (que entren en horas en el *sprint*) será la que más interese al cliente.

Una vez tenemos planificadas las tareas las distribuimos. Es el jefe de proyecto quien establece quien realiza cada tarea, dependiendo del *sprint*. Si es un *sprint* en el que el equipo va a estar ajustado temporalmente, se asignan las tareas a quienes mayores conocimientos tienen de estas. Si es un *sprint* más holgado se asignan a personas que no lo conocen, para que todos los miembros del equipo tengan conocimientos de todos los aspectos del proyecto. Esta asignación es solo temporal, debido a que el grupo de trabajo es nuevo, pues se pretende que sean los propios miembros del equipo los que repartan las tareas.

Tras la reunión de planificación el jefe del proyecto envía las tareas a las que se compromete el equipo al cliente y este acepta la planificación. Puede darse el caso de que hayamos dejado fuera una tarea urgente para el cliente, el cual puede solicitar cambiar la planificación. Este hecho no suele suceder ya que normalmente el cliente especifica correctamente las prioridades.

A mitades de *sprint* se le envía un correo informando de la situación del *sprint*, si se ha atascado en alguna tarea, si el equipo se va a desviar de la ruta tratada, si las tareas a las que se había comprometido no se van a poder finalizar a tiempo... Para que el cliente cuando llegue a la reunión de Review tenga unos mínimos conocimientos sobre cómo ha ido el trabajo en estas semanas.

5.2.3 Reunión diaria (*daily*)

Se trata de una reunión muy breve, de unos 15 minutos en la que cada miembro del equipo informa sobre el estado de la tarea que está realizando. Esta reunión se realiza cada día, el primer día del *sprint* se prepara la pizarra en la que aparece cada tarea con las horas totales estimadas y con el o los miembros del equipo que la van a realizar. A lo largo que transcurre el *sprint*, se van actualizando las horas que faltan para desarrollar cada tarea. Si una tarea ha costado menos de lo estimado, se apuntan las horas sobrantes en verde y si una tarea ha costado más de lo estimado en rojo. De esta forma es muy fácil ver el estado del *sprint*, pues basta con restar a las verdes las rojas y si sale negativo es que nos han costado en total todas las tareas más de lo estimado.



Ilustración 13 Fotografía de la pizarra Scrum

5.2.4 Reunión review

Se realiza al acabar el periodo planificado de dos semanas. En esta reunión participan los clientes, los desarrolladores y los jefes. Comienza el gestor de proyecto indicando que tareas de la planificación se han cumplido y que tareas no, argumentando las segundas. A continuación el responsable de proyecto explica en qué se ha materializado cada una de las tareas solicitadas por el usuario, respaldado por el equipo, que normalmente realiza explicaciones más técnicas o aclaraciones a las dudas del cliente.

5.2.5 Reunión retrospectiva

Es la última reunión de cada *sprint* en la que participan los desarrolladores, el gestor de producto y el responsable del proyecto. En esta reunión se establecen posibles mejoras para realizar en futuros *sprints*. Estas mejoras pueden ser tanto técnicas, como de planificación. En la siguiente tabla, ilustración 14 muestro algunas de las mejoras que hemos ido aplicando al trabajo diario, durante los últimos *sprints*.

Fecha	Issue	Mejora
15-03-17	Se han encontrado situaciones donde no se sabía que alcance quedaba dentro de una petición, o donde el título no reflejaba su contenido.	Redactar títulos que indiquen realmente lo que se solicita. Redactar descripciones que describan correctamente lo que se solicita
15-03-17	Se han realizado despliegues sin la documentación mínima exigida, por ejemplo, con DLyF desactualizados	Completar documentación mínima antes de desplegar.
15-03-17	Se han encontrado peticiones asignadas a dos trabajos evolutivos	Si una petición se va a desarrollar en dos versiones, dividir la petición en 2
15-03-17	Surgen problemas en despliegues debidos a merges mal resueltos en desarrollo	Ejecución diaria automática de carga en desarrollo para detectar problemas (unos serán esperados, pero nos interesan los no esperados)
15-03-17	Despliegues muy costosos, y conocimiento centralizado en una persona	Para cada desarrollo se prepara la parte del despliegue correspondiente
06-04-17	Tenemos impresión de que Infraestructuras piensa que todo lo de ER es "para ahora"	Ser más previsibles. Notificar con antelación, en cuanto conozcamos las necesidades para la que necesitaremos su ayuda
06-04-17	SPM a partir de las 13:30 provoca problemas para la asistencia de todo el EqDes	Evitar esa franja horaria
11-05-17	Tras la construcción se comprueba que no se ha ajustado a los requerimientos	Validar por parte de GProd los diseños antes de la construcción

Ilustración 14 Tabla de mejoras, reunión retrospectiva

5.3 Análisis de la metodología

Creo que esta metodología se adapta muy bien al trabajo que realizamos y que sin ella sería mucho más complejo comprender las necesidades del cliente y planificar correctamente al equipo de desarrollo. Lo más importante es la flexibilidad que mostramos al cliente, para que este decida en cada momento en que quiere que el equipo se centre y además puede ir comprobándolo en reducidos intervalos de tiempo. Esta metodología también ayuda a que este cliente esté más comprometido con el proyecto, ya que cada dos semanas se le explica lo desarrollado y nos transmite los nuevos requisitos.

6.Desarrollo

En este capítulo, el capítulo principal, explico cómo se ha desarrollado todo el Sistema de Explotación siguiendo el diseño del capítulo cuatro. Este desarrollo ha supuesto numerosas pruebas de concepto, prueba y error, para llegar al desarrollo limpio y completo que explico a continuación. Pero este desarrollo, no es perfecto y se puede mejorar, de hecho se hará en futuras versiones, pues este es un proyecto de mejora continua.

6.1 Componentes de la solución

La solución que el equipo de BI hemos desarrollado, consiste en un sistema que permite la explotación de la información clínica y administrativa de Salud de todos los pacientes residentes o atendidos en la comunidad foral de Navarra. Los datos se almacenan en repositorios centralizados donde se integra la información procedente de los diferentes productos corporativos. Además de la integración de datos, nuestro Sistema de Explotación destaca por presentar la información en estructuras muy comprensibles y con un excelente rendimiento para las consultas.

El producto que hemos desarrollado tiene varios componentes, entre ellos los interfaces del SE para productos o usuarios finales, elementos de puesta en marcha del sistema y planificación de la puesta en marcha del sistema.

6.1.1 Interfaces del SE

Un interfaz del SE son las vistas de Tableau, destinadas a los usuarios finales. Permiten facilitar la consulta de información almacenada en el DM sin necesidad de conocimientos sobre realizar consultas en SQL, ni demasiados conocimientos informáticos, ya que los usuarios finales suelen ser médicos, encargados de hospitales y directores del departamento de salud navarro. Podemos observar como ejemplo de las vistas de Tableau, el siguiente cuadro de mandos del SNS-O, en la ilustración 15.

Modelos de datos del Data Mart, permiten que el usuario final, utilizando diferentes herramientas pueda analizar y trabajar la información almacenada en el Data Mart del SE Poblacional.

Datos de auditoría, construimos una base de datos única y exclusivamente para almacenar los datos de auditoría. Permite que el usuario final, jefes del departamento de salud de Navarra y responsables de los distintos productos que utilizamos como fuentes de datos, consulten esta BBDD relacional para obtener información sobre la carga de los modelos de datos del DM. Permite al usuario tener conocimiento sobre el estado de los datos finales, pero sobretodo interesa el estado de cómo recibimos los datos de las fuentes. Para que estas puedan poco a poco mejorar sus productos y la forma en la que recogen los datos.

Consultas de datos del DM, los destinatarios son los distintos productos de Gobierno de Navarra, a los cuales permitimos interaccionar con el SE para obtener datos para su propio consumo y uso.

BBDD de análisis, utilizadas por los usuarios finales para ayudar en el proceso de análisis de la información almacenada en el Data Mart.



Ilustración 15 Cuadro de Mando en Tableau de SNS-O (Datos ficticios de 2100)

6.1.2 Elementos de puesta en marcha del sistema

Hemos realizado un *job* del agente de SQL Server que carga completamente el DM. Básicamente lo que hace es ejecutar diferentes paquetes *dtsx* para cargar todos los modelos de datos del DM del Sistema de Explotación.

Además, disponemos de otro *job*, similar al anterior, pero para ejecutar el proceso de Auditoría.

En cada paquete de procesos de Visual estudio, generamos un proceso que es el INDEX y llama a todos los demás procesos, que por ejemplo en la extracción tenemos uno por tabla extraída.

En la siguiente ilustración, podemos observar muy fácilmente el estado de la ejecución de cada *Index*. Gracias a la interfaz que ofrece SQL Server, ilustración 16, se ve muy claramente si la carga ha sido exitosa o no. Además dentro de cada fallo, se indica en que proceso del *index* ha fallado y muestra el motivo del error.

Execution Information

12 Failed **2** Running **258** Succeeded **10** Others

ID	Status	Report	Folder Name	Project Name	Package Name	Start Time	End Time	Duration (sec)
56408	Running	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_Leire	0_V_Leire_INDEX.dtsx	09/06/2017 8:02:11		677.256
56407	Running	Overview All Messages Execution Performance	Cronicos - BI.desarrollo	LD_Cronicos	00_LD_Cronicos_EXEC.dtsx	09/06/2017 7:40:42		1966.642
56406	Failed	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_Leire	0_V_Leire_INDEX.dtsx	09/06/2017 7:12:15	09/06/2017 7:58:10	2755.008
56405	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_Lamia	0_V_Lamia_INDEX.dtsx	09/06/2017 7:05:31	09/06/2017 7:12:13	401.925
56404	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_Lakora	0_V_Lakora_INDEX.dtsx	09/06/2017 7:03:39	09/06/2017 7:05:28	109.563
56403	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	Log	0_Log_INDEX.dtsx	09/06/2017 7:00:01	09/06/2017 7:00:11	10.171
56402	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_HCI	0_V_HCI_INDEX.dtsx	09/06/2017 6:14:17	09/06/2017 7:03:36	2958.338
56401	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_FARHO	0_V_FARHO_INDEX.dtsx	09/06/2017 6:12:00	09/06/2017 6:14:15	134.266
56400	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_CENTSENSO	0_V_Centenso_INDEX.dtsx	09/06/2017 6:11:43	09/06/2017 6:11:58	15
56399	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_Atenea	0_V_Atenea_INDEX.dtsx	09/06/2017 6:11:10	09/06/2017 6:11:40	30.172
56398	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	V_Configuracion	0_V_Configuracion_INDEX.dtsx	09/06/2017 6:11:03	09/06/2017 6:11:07	3.546
56397	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	M_Maestros	0_M_Maestros_INDEX.dtsx	09/06/2017 6:10:48	09/06/2017 6:11:01	12.734
56396	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	E_RPS	0_E_RPS_INDEX.dtsx	09/06/2017 6:09:03	09/06/2017 6:10:43	100.626
56395	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	E_UsuariosNavarra	0_E_UsuariosNavarra_INDEX.dtsx	09/06/2017 6:08:45	09/06/2017 6:08:58	13.001
56394	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	E_OtrasFuentes	0_E_OtrasFuentes_INDEX.dtsx	09/06/2017 6:08:38	09/06/2017 6:08:42	4.156
56393	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	E_Leire	0_E_Leire_INDEX.dtsx	09/06/2017 5:47:05	09/06/2017 6:08:35	1289.871
56392	Succeeded	Overview All Messages Execution Performance	BI.SEPoblacional.Des.arrollo	E_Lamia	0_E_Lamia_INDEX.dtsx	09/06/2017 5:35:06	09/06/2017 5:47:04	718.818

Ilustración 16 Estado de Ejecución

6.1.3 Planificación de puesta en marcha del sistema

Se realiza una carga completa del DM, que se ejecuta todos los domingos a las 8:00, cuando no afecta a los usuarios finales. Y otra carga del proceso de auditoría los lunes a las 6:00. El proceso de auditoría se carga posterior al del DM, porque es necesario para ejecutar el de auditoría que el del DM este cargado, ya que hace el proceso de auditoría es extraer datos de cómo ha ido la carga. Se ejecutan el domingo porque es el horario en el que no afectan a los usuarios finales y además el lunes si ha habido algún fallo, se realiza la corrección poco después de que acaben las ejecuciones.

6.2 Realización del desarrollo

Una vez realizado el análisis y diseño del sistema de explotación, pasamos al desarrollo. En este apartado explicaré en detalle cómo se estructura todo el sistema, las bases de datos que hemos creado, las tablas, etc.

6.2.1 Bases de datos de las réplicas

Como podemos observar en la Ilustración 17, este proyecto extrae datos de muchas fuentes. En esta captura no aparecen todas las bases de datos de las que extraemos información, pero sirve para ilustrar un poco todo este proyecto. Las bases de datos con las que trabajamos nosotros son las que tienen como sufijo _EXPLOTBI y podemos apreciar claramente, en el nombre, a que producto de los de Gobierno de Navarra corresponde. Por ejemplo AteneaGestion_EXPLOTBI es una réplica de la base de datos de Atenea de Gestión.

En la imagen también podemos apreciar como cada producto realiza las réplicas de formas diferentes. Los hay algunos que la realizan en una única base de datos y otros que en varias. Además hay algunos que nos aportan sus históricos y otros que no.

Aunque en esta ilustración solo observamos los datos que extraemos de fuentes que utilizan bases de datos (OLTP) como ya he explicado con anterioridad, también extraemos datos de productos en hojas de Excel por ejemplo.

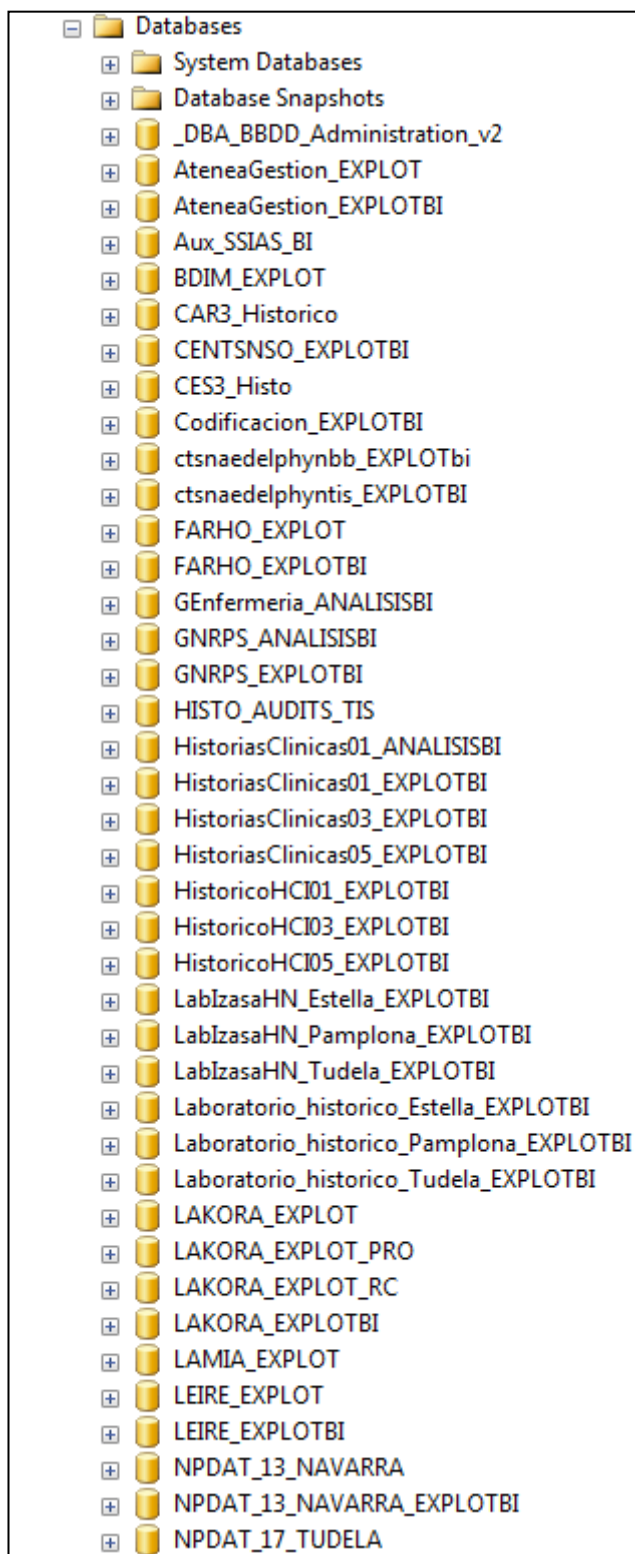


Ilustración 17 BBDD de las réplicas

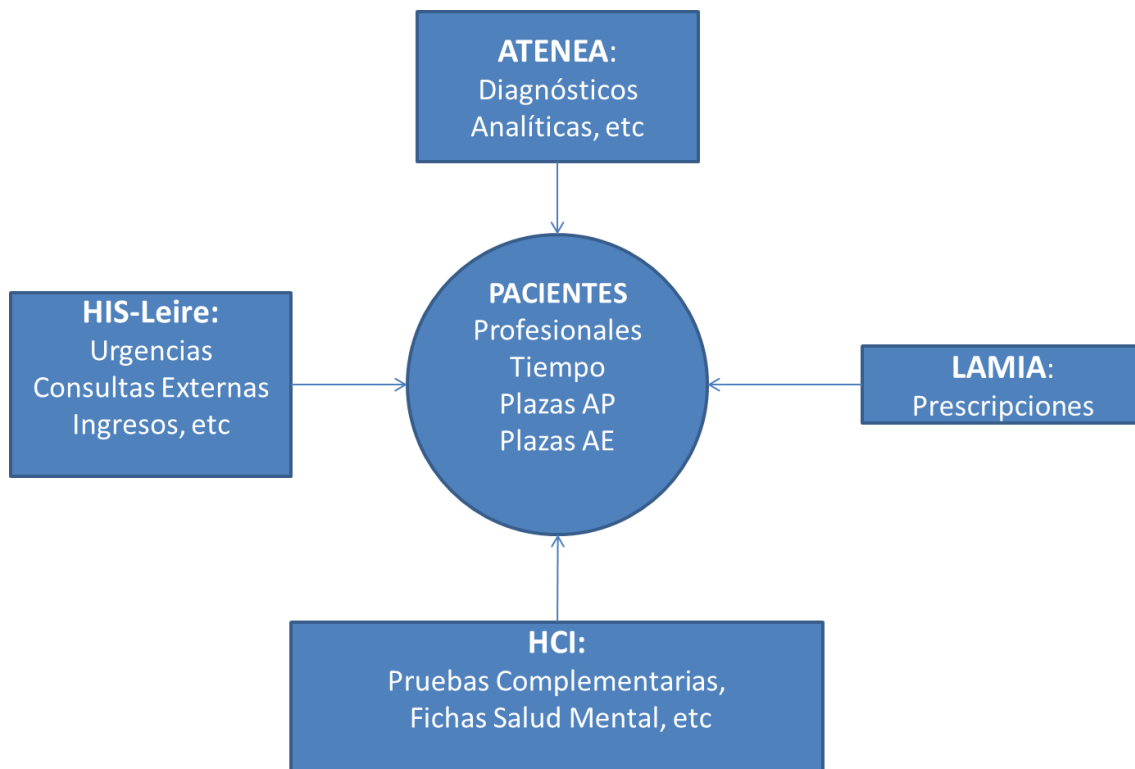


Ilustración 18 Esquema de los productos replicados

A continuación voy a explicar brevemente los productos a partir de los cuales obtenemos los datos, que transformamos en información. Como observamos en el esquema, ilustración 18, tienen especial importancia los datos obtenidos de Atenea, Leire, Lamia y HCI. Los cuatro son productos de Gobierno de Navarra. Además de otros productos como Lakora, y Observa que complementa a Lamia.

Atenea

Atenea se emplea para la gestión de Atención Primaria. Permite almacenar los DGP (Datos Generales Paciente) que se toman durante una cita en AP, el diagnóstico, las analíticas y en general datos de la cita. Es un producto con muchas peculiaridades. Por ejemplo tienen las bases de datos, repetidas en varios lugares, para cada uno de los sitios donde se utilizaba. (Los hospitales más grandes de Navarra), esto se debe a que es una aplicación muy antigua, que se desarrolló cuando las comunicaciones no eran lo que son ahora y no permitían fácilmente el traspaso de grandes cantidades de datos, de modo que creaban una base de datos en cada centro. Actualmente, se siguen empleando estas bases de datos separadas por centros, aunque tecnológicamente no son la mejor opción. Ya que es mucho más eficiente tener una gran base de datos en la que se almacenen los datos de los diferentes centros, y que cada centro envíe su información recabada. No aparece en la ilustración de las réplicas, pero solamente de Atenea genera 64 bases de datos, las cuales tienen exactamente la misma estructura.

Lamia

Lamia es la solución Informática para la gestión de las prescripciones, visados y dispensaciones electrónicas. Consta de los siguientes módulos: Administración: gestiona parámetros de configuración y usuarios del resto de módulos. Usado por los administradores de la aplicación que trabajan en SSIAS. Prescripción: engloba todos los procesos relacionados con la prescripción de medicamentos. Usado por médicos y directores de unidades de Atención Primaria y Atención Especializada. Dispensación: da soporte a los procesos de dispensación realizados en una oficina de farmacia. La aplicación la emplean los farmacéuticos y es capaz de integrarse con 10 programas propietarios que emplean las farmacias. Visado: permite gestionar los procesos relativos a la autorización o denegación de prescripciones de medicamentos que requieren de una validación previa a la dispensación, por varios motivos, y que es realizada por los profesionales de la oficina de inspección y visados de medicamentos (revisores e inspectores) Facturación: dota al personal del servicio de Prestaciones Farmacéutica de mecanismos de control para el proceso de facturación electrónica, que permiten garantizar que lo pagado por GN a las farmacias se corresponde con lo que realmente dispensan y facturan, cumpliendo el Acuerdo Marco suscrito por GN y COFNA.

Observatorio de la preinscripción (Observa)

Observa es un producto que se integra con Lamia y permite acceder a los tratamientos de un paciente y la comunicación entre los FAP (farmacéuticos de atención primaria) y los médicos de AP. Esto supone una gran ayuda, para conocer no solo si a un paciente se le ha prescrito un medicamento, sino también si ha acudido a recogerlo. De este modo nuestros análisis podrán mejorar, de modo que no se contará que un paciente está tomando un medicamento porque se lo hayan prescrito, sino se contará que lo toma si se lo ha comprado en la farmacia. Este dato tampoco será fiable al cien por cien, pero si se ajustará más a la realidad del anterior.

HCI

Sistema clínico-asistencial de Atención Especializada. La emplean los especialistas, para cada paciente abren una ficha donde almacenan los datos del paciente. Por ejemplo en salud mental, existe una ficha general que solicita datos como diagnóstico primario, secundario y terciario... Existen fichas de diferentes especialidades y también fichas para introducir los datos de la realización de una prueba.

Leire

Herramienta para la gestión de la información hospitalaria y consultas externas especializadas usada por los administrativos del SNS-O. Permite la apertura, búsqueda y modificación de historias clínicas. Su principal función, en urgencias es registrar las urgencias, el motivo y la fecha, además de registrar los ingresos. De cada ingreso conoceremos todos sus movimientos, cambios de habitación, traslado de una especialidad a otra, fecha de alta... Para facilitar todas estas funciones mencionadas, genera numerosos informes con esta información.

Lakora

Aunque no aparezca en el esquema, Lakora también es un producto que nos aporta muchos datos. Respecto a los médicos almacena las plazas, el CIAS, los partidos médicos... También almacena datos de los pacientes, en esta aplicación se gestiona la tarjeta sanitaria, donde vive cada paciente, cuál es su centro de salud, su zona básica, etc.

6.2.2 Bases de Datos del modelo de BI

En la siguiente ilustración, la número 19, observamos las bases de datos que empleamos en el modelo relacional y que ya he mencionado anteriormente. En la siguiente página explico cómo y para qué generamos cada una de las bases de datos que aparecen remarcadas.

La construcción comienza en el SA_RIN que como su nombre indica es un repositorio normalizado *Staging Area*, que sirve de apoyo para la extracción y la transformación.

A partir de este repositorio, se genera el RIN y el DM_Poblacional. El RIN es un repositorio normalizado que empleamos de ayuda para algunas tablas de hechos y la historificación.

Por último el DM_Poblacional es donde se almacenan las dimensiones y tablas de hechos del Data Mart.

A continuación se explicará detalladamente como se crean las tablas de cada de la bases de datos y los procesos que generamos para crearlas.

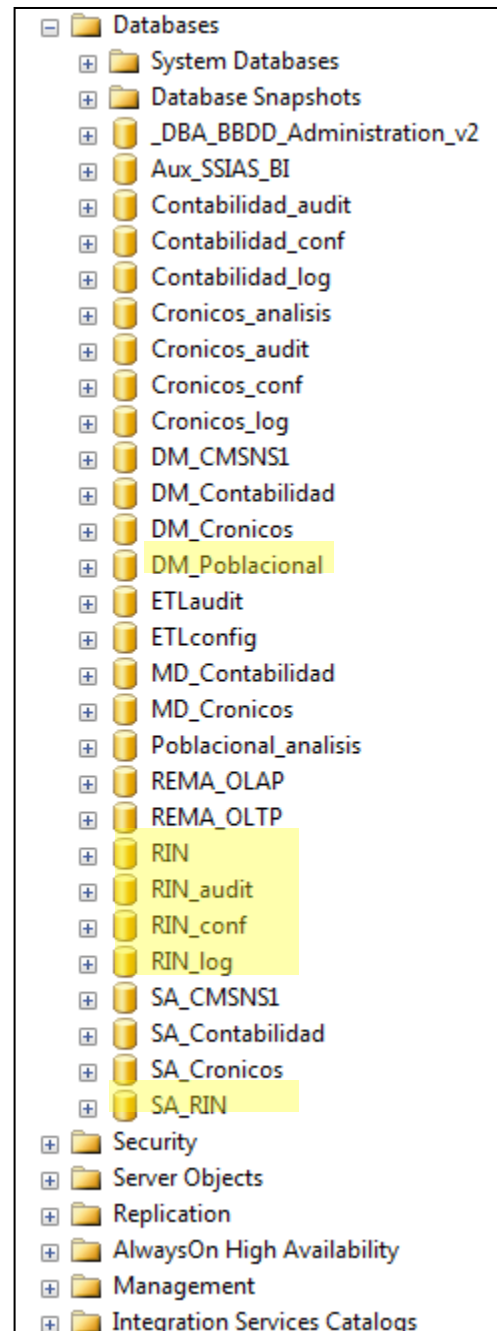


Ilustración 19 Bases de Datos modelo BI

6.2.2.1 Extracción de datos (SA_RIN)

En este apartado se explica la metodología que seguimos para realizar la extracción de los datos de las tablas fuente. Las tablas de origen se extraen de las bases de datos de las réplicas y las tablas destino, donde extraemos los datos, se crean en la base de datos SA_RIN.

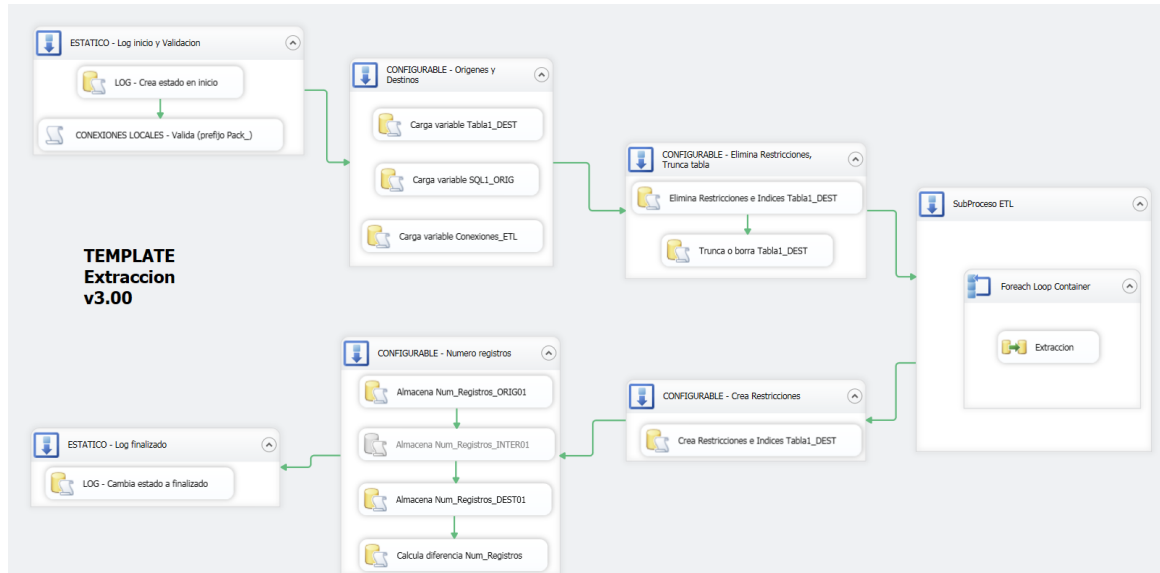


Ilustración 20 Template en Visual Studio para la Extracción de datos

En la ilustración 20 contiene el template de Extracción completo, podemos observar varios contenedores. Los que están denominados como configurables significa que hay que modificar algo en ellos cuando lo reutilicemos, mientras que los que se denominan como estáticos, permanecen intactos. Este template, actúa sobre varias bases de datos. Por ello necesitaremos las siguientes conexiones: una conexión a las bases de datos de las réplicas para obtener la tabla origen, otra conexión al SA_RIN donde se almacenará la tabla destino, y dos conexiones más a RIN_log y RIN_conf donde se almacena información sobre el estado del template. Se explicarán más adelante, en sus correspondientes apartados.

Además a la derecha de la ilustración podemos observar una caja llamada Extracción. Esta caja contiene el data flow que se muestra a continuación, en la ilustración 21.

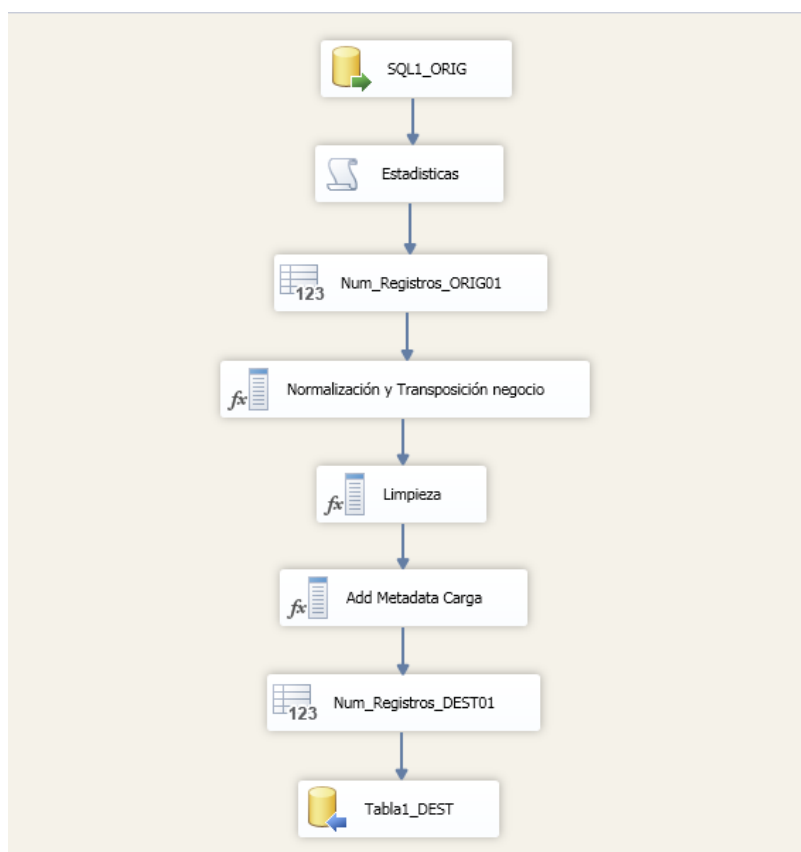


Ilustración 21 Data Flow del proceso de Extracción de datos

En este proceso tratamos los datos que recibimos de las tablas de las réplicas. Por ejemplo si tenemos varias tablas que almacenan los mismos datos pero de distintos centros hospitalarios, como es el caso de Atenea, los juntamos en una sola tabla. En esta tabla, renombrada con el prefijo E_ tendremos todos los campos originales con sus tipos originales y una copia de cada uno renombrado con el prefijo Z_. A estas copias se les asigna un nuevo nombre que sea representativo del contenido del campo y un tipo acorde. En muchos casos los nombres de los campos de las tablas de origen no son representativos. Observar el ejemplo que aparece en la siguiente ilustración, donde el campo Cotifarm pasa a denominarse Z_Cod_Cobertura_Farmaceutica. Además los tipos de los campos de entrada suelen ser *varchar*, se correspondan con fechas, con números o con cualquier otro tipo de valores.

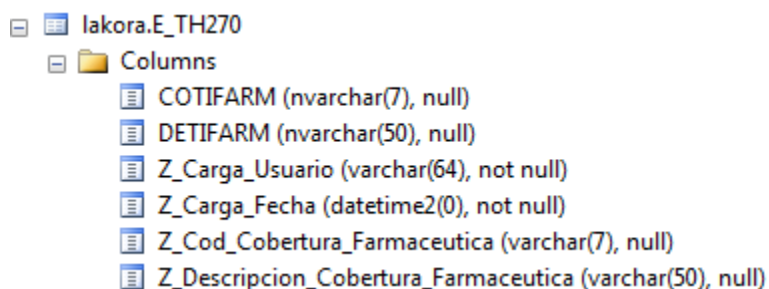


Ilustración 22 Ejemplo tabla extraída de Lakora, con nombres poco representativos

La primera caja denominada SQL1_ORIG, es un *Source Assistant* en el cual realizamos la consulta SQL para obtener los datos de la tabla origen de la que queremos extraer los datos. Estas tablas se encuentran en las bases de datos de las réplicas. En este ejemplo, el SQL1_ORIG contiene la siguiente consulta:

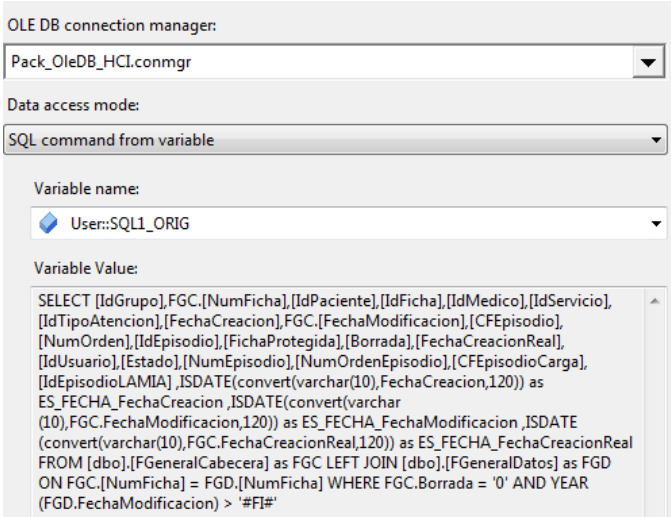


Ilustración 23 Extracción de datos. Consulta SQL1_ORIG

Podemos ver que es una consulta que extrae datos de varias tablas de un DB de las réplicas de HCI. Esta consulta tiene dos condiciones, una establece la fecha de los datos más antiguos y otra si estos datos son el último cambio y no se han sustituido por otros. Además genera dos campos nuevos con el prefijo ES_FECHA_ que posteriormente se emplean para comprobar que los datos que recibimos en campos que conocemos que son fechas lo son. Esta comprobación es necesaria porque los datos de origen provienen de un campo de tipo *varchar* y no de uno de tipo fecha.

Las cajas Estadísticas y el *row-count Num_Registros_Origen* son para recopilar información sobre la extracción. Número de filas en la tabla origen, tiempo en extraerlo... Se explicará más detalladamente en el apartado del *RIN_log*.

La caja Normalización y Transposición negocio es un apartado donde tratamos todos los datos que recibimos según las indicaciones del cliente, se muestra en la ilustración 24.

Derived Column Name	Derived Column	Expression
Z_Num_Orden	<add as new column>	(DT_I4)NumOrden
Z_Id_Episodio	<add as new column>	(DT_STR,16,1252)IdEpisodio
Z_Ficha_Protegida	<add as new column>	(DT_I4)FichaProtegida
Z_Borrada	<add as new column>	(DT_I4)Borrada
Z_Fecha_Hora_Creacion_...	<add as new column>	ISNULL(FechaCreacionReal) ? (DT_DBTIMESTAMP)@[User::Fecha_Timestamp_Defecto] : ES_FECHA_FechaCreacionReal == 0 ?
Z_Fecha_Creacion_Real	<add as new column>	ISNULL(FechaCreacionReal) ? (DT_DBDATE)@[User::Fecha_Date_Defecto] : ES_FECHA_FechaCreacionReal == 0 ? (DT_DBDATE
Z_Hora_Creacion_Real	<add as new column>	ISNULL(FechaCreacionReal) ? (DT_DBTIME)@[User::Hora_Time_Defecto] : ES_FECHA_FechaCreacionReal == 0 ? (DT_DBTIME)
Z_Id_Usuario	<add as new column>	(DT_STR,8,1252)IdUsuario
Z_Estado	<add as new column>	(DT_STR,2,1252)Estado
Z_Num_Episodio	<add as new column>	(DT_I4)NumEpisodio
Z_Num_Orden_Episodio	<add as new column>	(DT_I4)NumOrdenEpisodio
Z_CF_Episodio_Carga	<add as new column>	(DT_STR,2,1252)CFEpisodioCarga
Z_Id_Episodio_LAMIA	<add as new column>	(DT_STR,30,1252)IdEpisodioLAMIA
Z_Id_Paciente	<add as new column>	(DT_STR,20,1252)('PLR#' + RIGHT(REPLICATE('0',12) + TRIM((DT_STR,20,1252)IdPaciente),12))

Ilustración 24 Extracción de datos. Normalización y transposición

Para las fechas empleamos el nuevo campo generado en SQL1_ORIG y en caso de que el campo no sea una fecha correcta, le asignamos una fecha por defecto “9999-12-31” que sería la correspondiente a dato desconocido o erróneo.

Sobre otros campos simplemente se realiza una conversión del tipo del campo.

También hay algunos campos que creamos nosotros para futuros cruzamientos entre datos de diferentes tablas y de diferentes fuentes. Por ejemplo el campo *Z_Id_Paciente*, que se genera de la siguiente forma:



P: inicial del contenido del campo. En este caso Paciente = P

LR: las dos primeras iniciales de la fuente de la que provienen. En este caso HCI no da identificadores propios a sus pacientes, sino que emplea los de Leire de modo que su *Z_Id_Paciente* se forma como si fuera de Leire, Leire = LR.

#: Se emplea como separador

IdPaciente: en este caso el campo con el que formamos el identificador se llama *IdPaciente*, hay uno específico para cada fuente de datos. Por ejemplo en HCI es *IdPaciente*, pero en

En la caja de Limpieza realizamos un tratamiento sobre todos los nuevos campos eliminando posibles valores nulos o vacíos y los sustituimos -1, vemos el contenido en la ilustración 25.

Derived Column Name	Derived Column	Expression
Z_Num_Ficha	Replace 'Z_Num_Ficha'	ISNULL(Z_Num_Ficha) ? -1 : Z_Num_Ficha
Z_Id_Grupo	Replace 'Z_Id_Grupo'	ISNULL(Z_Id_Grupo) ? -1 : Z_Id_Grupo
Z_Id_Ficha	Replace 'Z_Id_Ficha'	ISNULL(Z_Id_Ficha) ? -1 : Z_Id_Ficha
Z_Num_Colegiado	Replace 'Z_Num_Colegiado'	TRIM(Z_Num_Colegiado) == "" ISNULL(Z_Num_Colegiado) ? "-1" : TRIM(Z_Num_Colegiado)
Z_Id_Servicio	Replace 'Z_Id_Servicio'	ISNULL(Z_Id_Servicio) ? -1 : Z_Id_Servicio
Z_Id_Tipo_Atencion	Replace 'Z_Id_Tipo_Atencion'	ISNULL(Z_Id_Tipo_Atencion) ? -1 : Z_Id_Tipo_Atencion
Z_CF_Episodio	Replace 'Z_CF_Episodio'	TRIM(Z_CF_Episodio) == "" ISNULL(Z_CF_Episodio) ? "-1" : TRIM(Z_CF_Episodio)
Z_Num_Orden	Replace 'Z_Num_Orden'	ISNULL(Z_Num_Orden) ? -1 : Z_Num_Orden
Z_Id_Episodio	Replace 'Z_Id_Episodio'	TRIM(Z_Id_Episodio) == "" ISNULL(Z_Id_Episodio) ? "-1" : TRIM(Z_Id_Episodio)
Z_Ficha_Protegida	Replace 'Z_Ficha_Protegida'	ISNULL(Z_Ficha_Protegida) ? -1 : Z_Ficha_Protegida
Z_Borrada	Replace 'Z_Borrada'	ISNULL(Z_Borrada) ? -1 : Z_Borrada
Z_Id_Usuario	Replace 'Z_Id_Usuario'	TRIM(Z_Id_Usuario) == "" ISNULL(Z_Id_Usuario) ? "-1" : TRIM(Z_Id_Usuario)
Z_Estado	Replace 'Z_Estado'	TRIM(Z_Estado) == "" ISNULL(Z_Estado) ? "-1" : TRIM(Z_Estado)
Z_Num_Episodio	Replace 'Z_Num_Episodio'	ISNULL(Z_Num_Episodio) ? -1 : Z_Num_Episodio
Z_Num_Orden_Episodio	Replace 'Z_Num_Orden_Episodio'	ISNULL(Z_Num_Orden_Episodio) ? -1 : Z_Num_Orden_Episodio
Z_CF_Episodio_Carga	Replace 'Z_CF_Episodio_Carga'	TRIM(Z_CF_Episodio_Carga) == "" ISNULL(Z_CF_Episodio_Carga) ? "-1" : TRIM(Z_CF_Episodio_Carga)
Z_Id_Episodio_LAMIA	Replace 'Z_Id_Episodio_LAMIA'	TRIM(Z_Id_Episodio_LAMIA) == "" ISNULL(Z_Id_Episodio_LAMIA) ? "-1" : TRIM(Z_Id_Episodio_LAMIA)
Z_Id_Paciente	Replace 'Z_Id_Paciente'	TRIM(Z_Id_Paciente) == "" ISNULL(Z_Id_Paciente) ? "-1" : TRIM(Z_Id_Paciente)
Z_NHC	Replace 'Z_NHC'	TRIM(Z_NHC) == "" ISNULL(Z_NHC) ? "-1" : TRIM(Z_NHC)

Ilustración 25 Extracción de datos. Limpieza

La caja que se reutiliza en todos los templates, es la de Add Metadata Carga. En esta caja añadimos dos nuevos campos Z_Fecha_Carga y Z_Usuario_Carga, son muy importantes para saber cuándo y quién ha realizado la carga de la tabla en cuestión. En un proyecto de BI, donde se trabaja con muchas tablas, es fundamental llevar un control estricto de las fechas en las que se cargan los datos para que estos sean consistentes. No se podría calcular ningún indicador sobre datos que no estén alineados temporalmente, pues obtendríamos resultados incoherentes. Si por ejemplo, queremos calcular el número de pacientes polimedicados y los datos que tenemos en el Data Mart se corresponden con distintas fechas. Puede ser que la tabla a partir de la cual calculamos si un paciente o no es polimedicado sea de 2016, mientras que la de los pacientes totales sea de 2009, como hay muchos más registros de pacientes en 2016, probablemente nos salga que hay más pacientes polimedicados que pacientes totales, lo

Derived Column Name	Derived Column	Expression
Z_Usuario_Carga	<add as new column>	(DT_STR,64,1252)@[System::UserName]
Z_Fecha_Carga	<add as new column>	@[System::StartTime]

cual es incorrecto. Un ejemplo del contenido de esta caja, se muestra en la ilustración 26.

Ilustración 26 Extracción de datos. Add Metadata

El row-count Num_Registros_DEST01 sirve para recopilar información sobre el número de filas que se van a introducir en la tabla destino de la extracción. Se explicará más detalladamente en el apartado del RIN_log.

Por último la Tabla1_DEST, selecciona la tabla sobre la que se quieren cargar los datos. Esta tabla se denominará con el prefijo E_ y se ubicará en la base de datos SA_RIN. En esta caja se realiza el mapeo que une cada campo de la tabla de origen y cada campo de la normalización con el campo de la tabla destino.

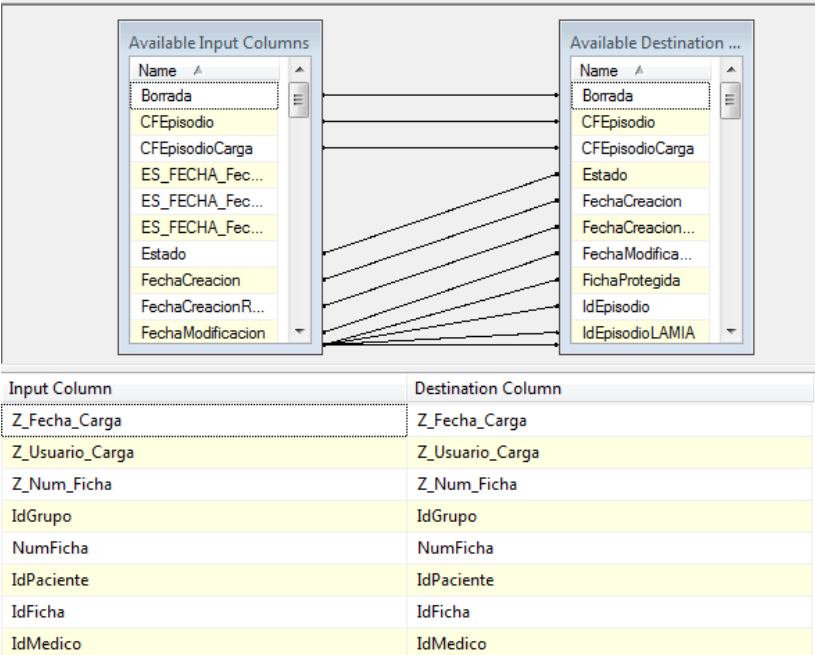
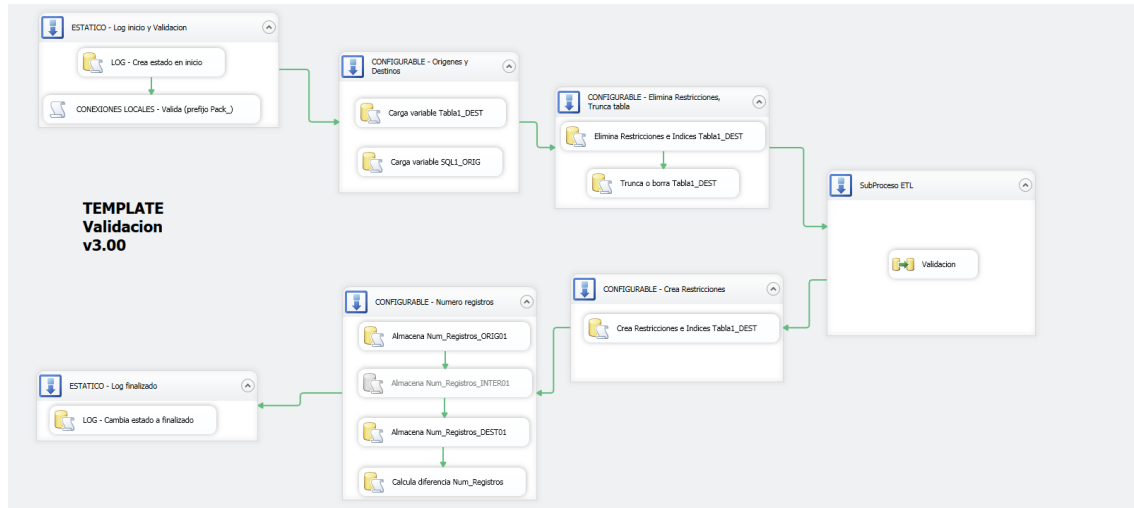


Ilustración 27 Extracción de datos. Tabla1_DEST

6.2.2.2 Validación de datos (SA_RIN)

En este apartado se explica la metodología que seguimos para realizar la validación de los datos de las tablas de extracción. En este proceso las tablas de origen y las de destino se



almacenan en la misma base de datos, el SA_RIN.

Ilustración 28 Template en Visual Studio para la Validación de datos

El template de la Extracción, como podemos ver en la ilustración 28, se reutiliza en la Validación. Sin embargo, el Data Flow es diferente. En este paso lo que hacemos es crear una nueva tabla con el prefijo V_ y el nombre de la tabla E_ de la que extrae los datos. En esta nueva tabla, que también se almacena en el SA_RIN, introduciremos solamente los campos generados en la extracción con el prefijo Z_. En esta tabla se mantiene el prefijo, de manera que los campos se llaman exactamente igual.

En cuanto a las conexiones necesarias en este template, necesitamos una conexión al SA_RIN donde se almacena la tabla origen (E_) y se almacenará la tabla destino (V_), y dos conexiones más a RIN_log y RIN_conf, que se explicarán en sus correspondientes apartados.

Además a la derecha de la ilustración podemos observar una caja llamada Validación. Esta caja contiene el data flow que se muestra a continuación, en la ilustración 29.

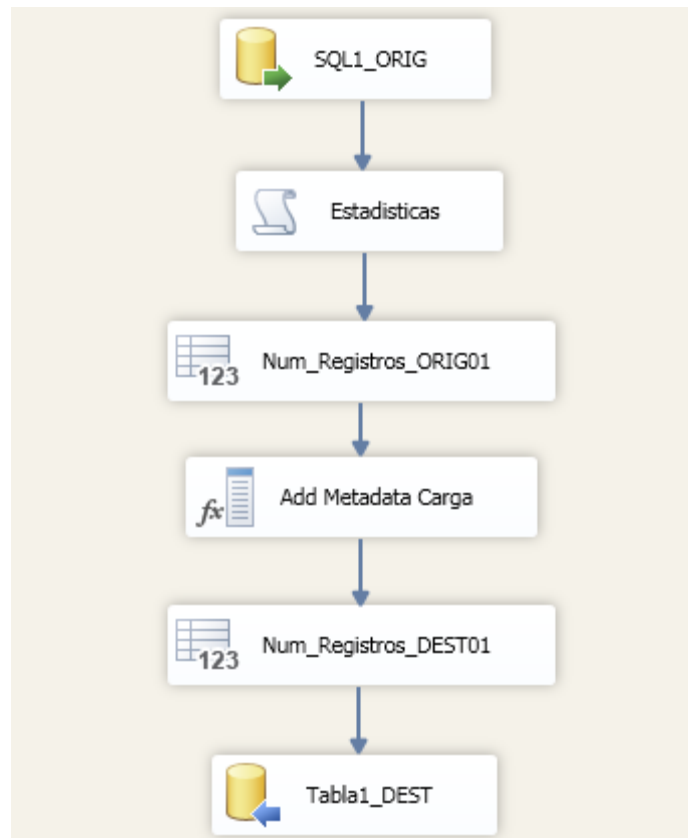


Ilustración 29 Data Flow del proceso de Validación de datos

Este data flow es más sencillo que el de extracción y reutiliza la caja Add Metadata Carga, mostrada en la ilustración 26. En este paso de la extracción a la validación, lo único que hacemos es reducir las tablas de la extracción quedándonos solo con los campos que hemos creado en la extracción.

6.2.2.3 Transformación de datos (SA_RIN)

En este apartado se explica la metodología que seguimos para realizar la transformación de los datos de las tablas de validación. En este proceso las tablas de origen y las de destino se almacenan en la misma base de datos, el SA_RIN.



Ilustración 30 Template en Visual Studio para la Transformación de datos

El template anterior, ilustración 30, es un template más sencillo que los de Extracción y Validación, porque no recoge tantos datos de estados de la carga y de la ejecución. Aun así, recoge datos, pero lo hace de manera más escueta. Esto se debe a que este template de Transformación es el de la versión anterior, mientras que los de Extracción y Validación son los de la última versión. Actualmente se está desarrollando el nuevo template para la Transformación, o mejor dicho la adaptación del template de Extracción para la Transformación. Ya que con el diseño del template de Extracción se buscó que pudiera ser reutilizable en todos los pasos del ETL, para que fueran más sencillos y comprensibles.

En resumen, el template de Transformación que explico es el de la versión anterior, porque el nuevo todavía no está disponible, algunas de las ventajas del nuevo, además de la amplia recogida de datos de estado y carga, es que las consultas y los nombres de las tablas se cargan desde una tabla de la base de datos y no desde el propio dtsx. Esto facilita los cambios que queramos realizar sobre las tablas de origen o destino, sin tener que modificar los procesos. En definitiva buscamos un template que se pueda reutilizar con el menor número de cambios. (Para reutilizar el template de Extracción de una tabla a otra, solamente hay que modificar los valores de dos variables, lo cual es un gran avance, ya que en el modelo anterior había que modificar dos consultas y cuatro variables como mínimo.)

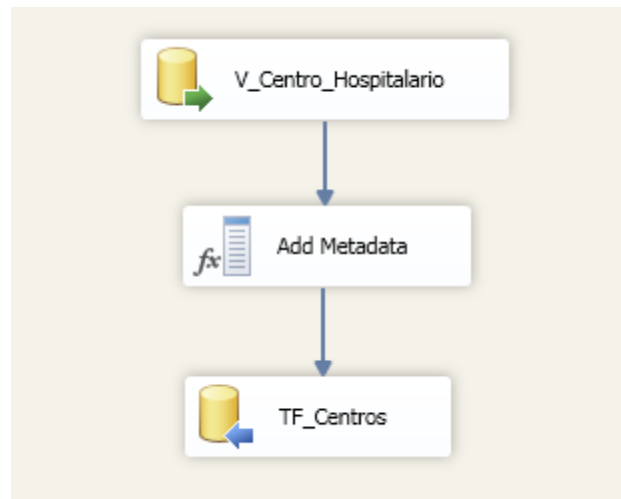


Ilustración 31 Data Flow del proceso de Transformación de datos

Este data flow, ilustración 31, es el más sencillo de todos y también reutiliza la caja Add Metadata. En este paso de la validación a la transformación, transformamos las tablas de validación en unas tablas preparadas para la generación de las dimensiones.

Derived Column Name	Derived Column	Expression
Usuario_Carga	<add as new column>	(DT_STR,64,1252)@[System::UserName]
Fecha_Carga	<add as new column>	@[System::StartTime]
Desc_Complejo	<add as new column>	(DT_STR,50,1252)(Z_Cod_Complejo == 1 ? "Complejo Hospitalario de Navarra" : Z_Cod_Complejo == 2 ?
Z_Cod_Centro	Replace 'Z_Cod_Centro'	(DT_I4)Z_Cod_Centro

Ilustración 32 Transformación de datos. Add Metadata

En la caja de Add Metadata, como se puede ver en la ilustración 32 no solo añadimos la fecha y el usuario de carga, sino que además realizamos un tratamiento sobre el código y la descripción de los campos de la tabla de validación. En este ejemplo, se corresponde con la tabla TF_Centros, que igual que casi todas las tablas TF tiene un campo que es el código y otro la descripción de este código. En esta tabla ya desaparecen los campos con el prefijo Z, para dar lugar a los nombres de los campos definitivos.

Podemos observar como creamos el nuevo campo Desc_Complejo en el que introducimos una descripción por cada código de complejo, Cod_Complejo. En esta caja se realizan tratamientos similares para los campos Cod_Centro y Desc_Centro.

Como aclaración a partir de esta tabla se genera la dimensión de centros. Cada centro hospitalario pertenece a un complejo hospitalario, de modo que con el código de un centro podremos saber el nombre del centro, el código del complejo y el nombre del complejo al que pertenece. Se muestra en la ilustración 33 el mapeo de los campos de la dimensión de la tabla TF_Centros.

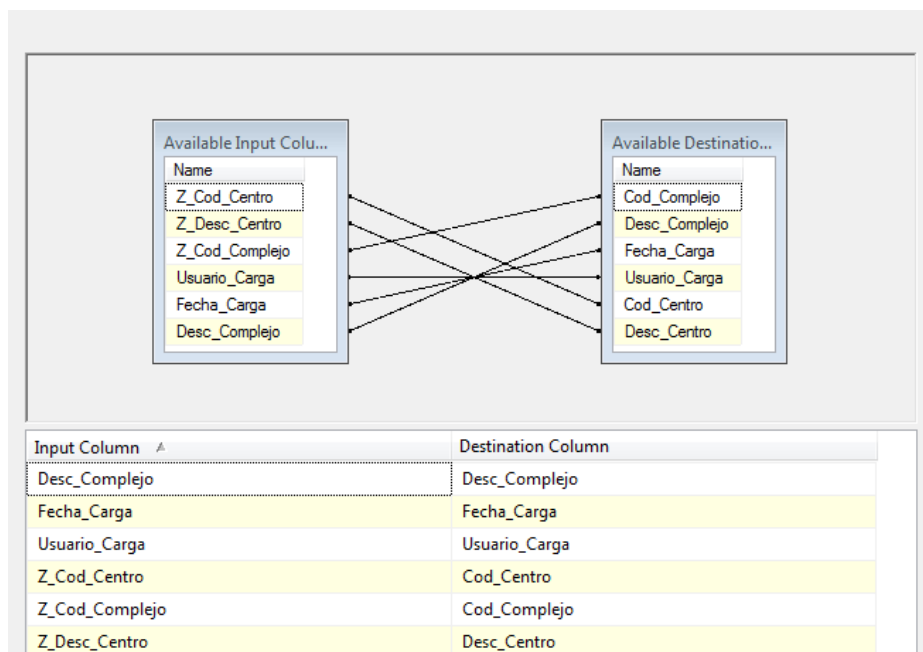


Ilustración 33 Transformación de datos. Tabla destino (TF_Centros)

6.2.2.4 Paso al Data Mart: dimensiones (DM_Poblacional)

En este apartado explicaré como pasamos de las tablas TF a las tablas de dimensiones que conformarán el Data Mart, resultado de todo este trabajo.

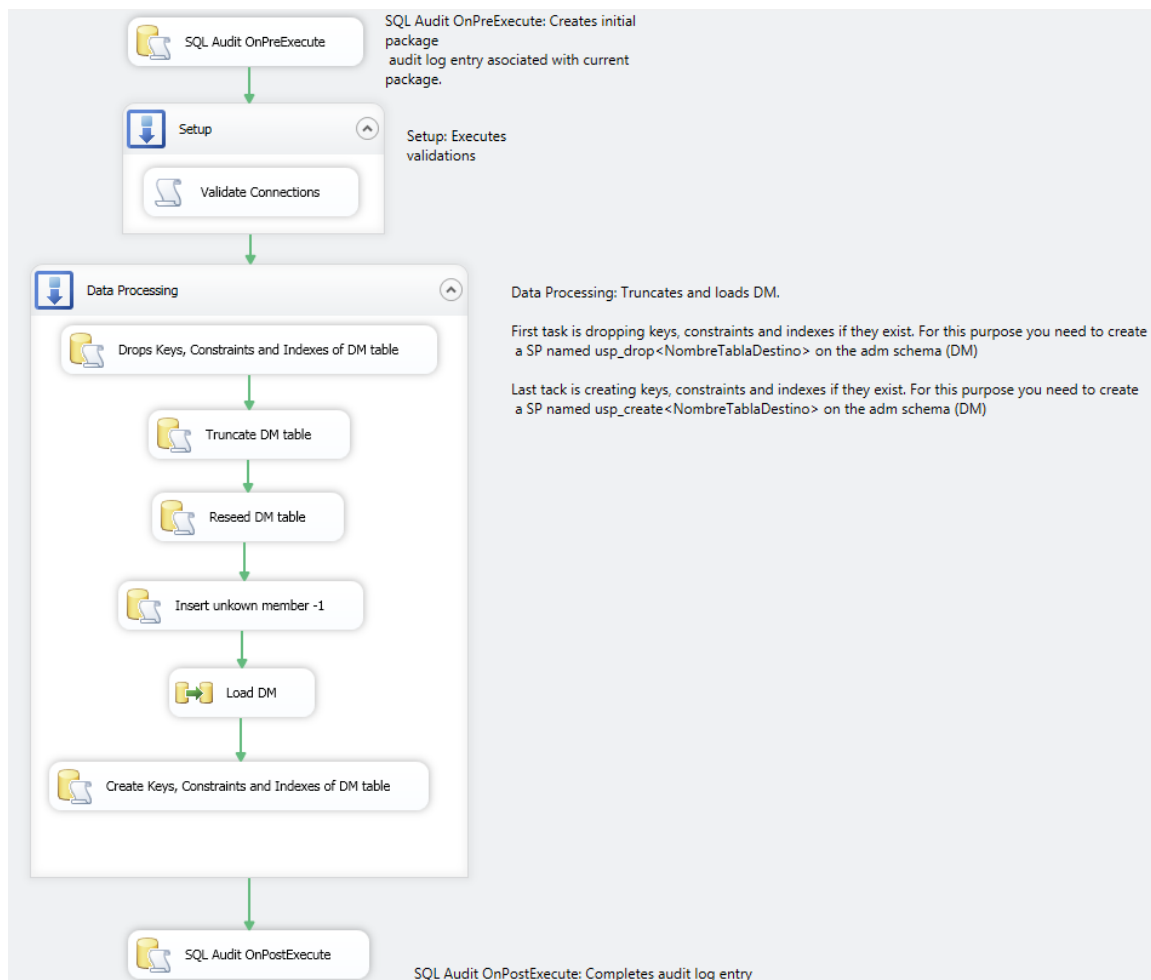


Ilustración 34 Template en Visual Studio para el paso al Data Mart de una dimensión

Como se puede observar claramente en la ilustración, este template es muy diferente a los explicados anteriormente y es porque realiza funciones muy diferentes también.

Dentro del contenedor denominado Data Processing, observamos un flujo compuesto por varias cajas, todos los pasos que menciono son sobre la tabla destino, en este caso una dimensión (Dim_Centros). En la primera, denominada Drop Keys... simplemente se eliminan las claves y los índices de la tabla de dimensiones que vamos a generar en este proceso. A continuación, en las dos siguientes cajas, se vacía la tabla con un truncate y se reinicia la semilla, es decir, el auto incremental del identificador se vuelve a poner en 1.

Dentro del Insert unknown... se inserta la primera fila de la dimensión. Esta fila es la asociada a desconocido, que tendrá en todos los campos los valores -1 o desconocido. Es muy importante la inserción de esta fila, pues cuando crucemos tablas de hechos con esta dimensión, si un

dato viene con un código que no conocemos, es decir, que no existe en la dimensión, se mapeará con el -1 y se considerará un valor desconocido, en lugar de perder el dato.

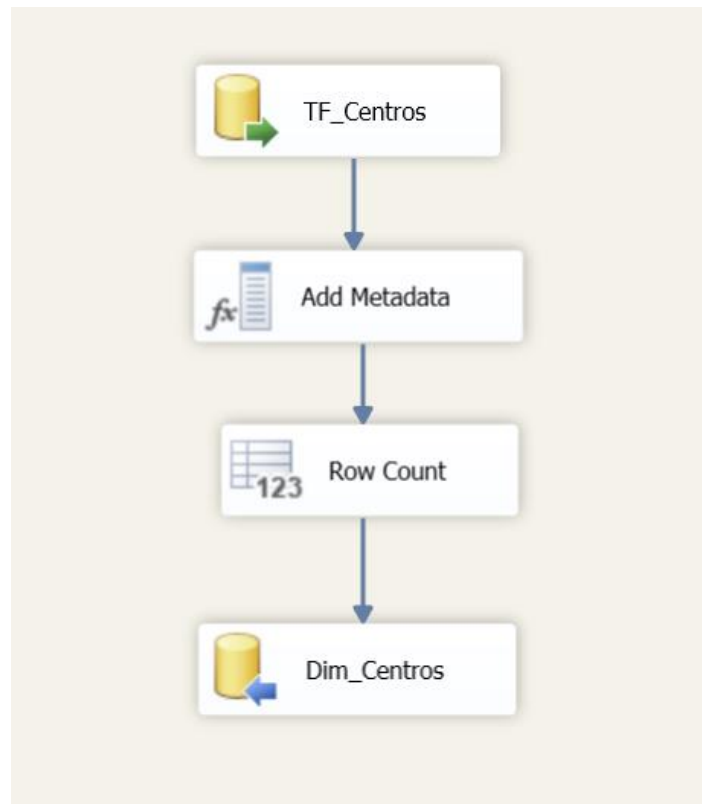


Ilustración 35 Data Flow del proceso de paso al Data Mart de una dimensión

Este data flow, mostrado en la ilustración 35, es muy sencillo. Reutiliza la caja Add Metadata y en la caja Dim_Centros se mapean los campos de la TF a la DIM. Como podemos observar en la siguiente captura, la SK_Centro que es la clave primaria de la dimensión que estamos generando, no se mapea. Esto se debe a que es un campo auto incremental.

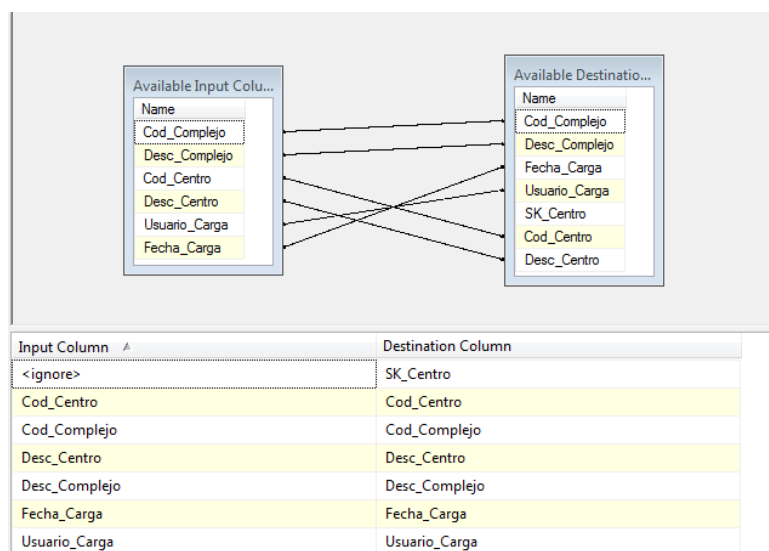


Ilustración 36 Tabla destino del proceso de paso al Data Mart de una dimensión

Tras todo este trabajo de procesamiento y tratamiento de datos logramos obtener las dimensiones. En la ilustración 36, se observa el mapeo de los campos tratados con los de la dimensión centros. Anteriormente he explicado la forma en la que el equipo desarrolla los procesos de ETL, que se corresponden de la siguiente forma:

- Extract: Extracción y Validación
- Transform: Transformación
- Load: Paso al Data Mart

En la siguiente ilustración muestro una de las dimensiones que forman nuestro Data Mart, la dimensión de centros que he utilizado para las explicaciones anteriores.

SK_Centro	Cod_Centro	Desc_Centro	Cod_Complejo	Desc_Complejo	Fecha_Carga	Usuario_Carga
1	-1	Desconocido	-1	Desconocido	2017-06-01 11:50:42	ADMON-CFNAVARRA\SVC_POB_Ueti_DES
2	1	Hospital de Navarra	1	Complejo Hospitalario de Navarra	2017-06-01 11:50:41	ADMON-CFNAVARRA\SVC_POB_Ueti_DES
3	2	Hospital Reina Sofía	2	Hospital Reina Sofía	2017-06-01 11:50:41	ADMON-CFNAVARRA\SVC_POB_Ueti_DES
4	3	Clínica Ubamin	1	Complejo Hospitalario de Navarra	2017-06-01 11:50:41	ADMON-CFNAVARRA\SVC_POB_Ueti_DES
5	4	Hospital García Orcoyen	3	Hospital García Orcoyen	2017-06-01 11:50:41	ADMON-CFNAVARRA\SVC_POB_Ueti_DES
6	5	Hospital Virgen del Camino	1	Complejo Hospitalario de Navarra	2017-06-01 11:50:41	ADMON-CFNAVARRA\SVC_POB_Ueti_DES
7	6	Centro San Francisco Javier	4	Salud Mental	2017-06-01 11:50:41	ADMON-CFNAVARRA\SVC_POB_Ueti_DES
8	7	Hospital a domicilio	1	Complejo Hospitalario de Navarra	2017-06-01 11:50:41	ADMON-CFNAVARRA\SVC_POB_Ueti_DES

Ilustración 37 Dimensión de centros

6.2.2.5 Paso al Data Mart: tablas de hechos (RIN y DM_Poblacional)

En este apartado se explica la metodología que seguimos para generar las tablas de hechos, a partir de las dimensiones y las tablas TF. De modo que para realizar las tablas de hechos se necesita que todas las dimensiones estén cargadas.

Una tabla de hechos es la tabla central de un modelo en estrella, un hecho es un indicador o medida de negocio, que posteriormente será lo que aparezca en el sistema de Business Intelligence.

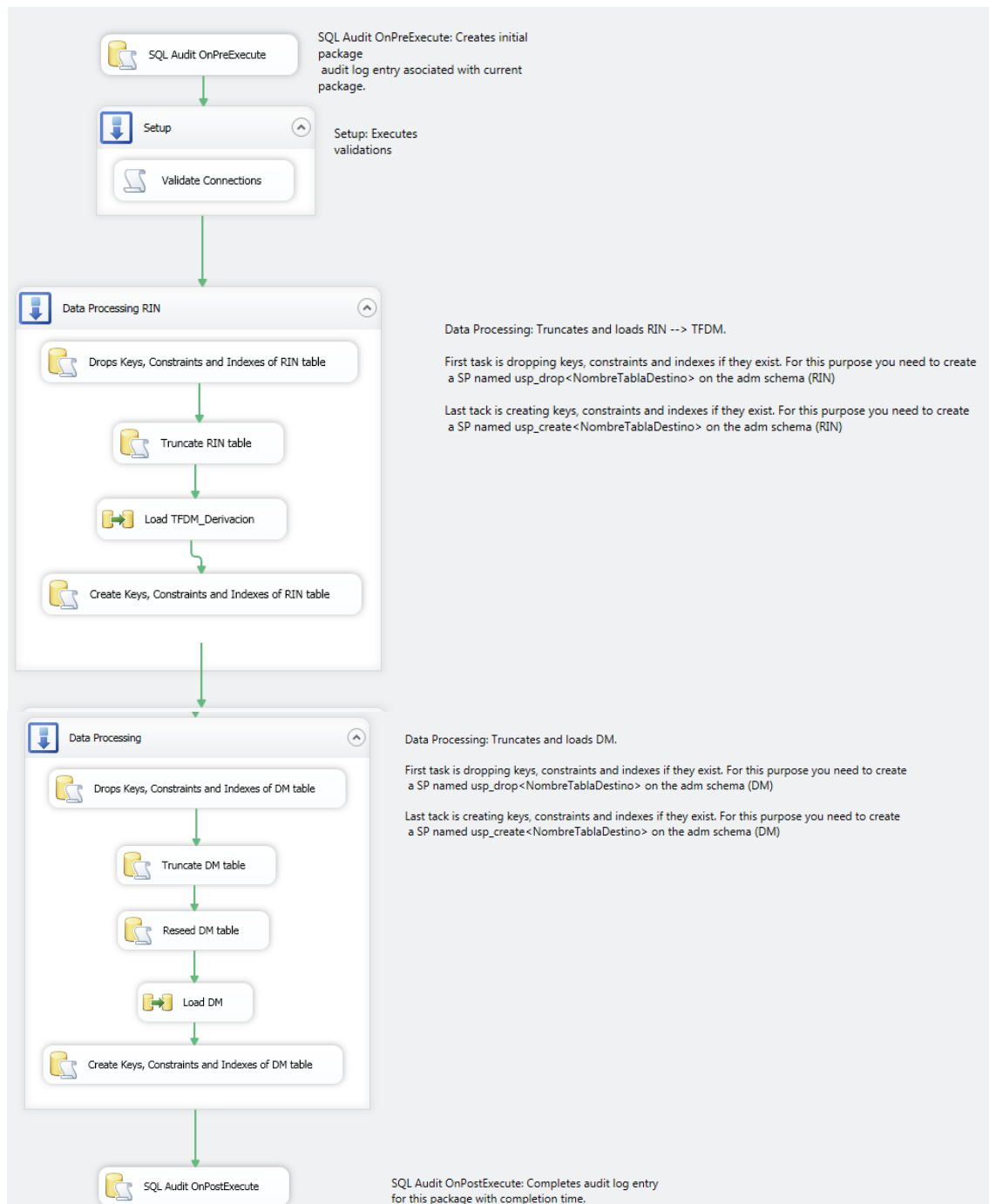


Ilustración 38 Template en Visual Studio para el paso al Data Mart de una tabla de hechos

En esta ilustración 38 podemos observar la forma que tiene el template del paso al DM de una tabla de hechos. Si lo comparamos con el de una dimensión, la primera parte es similar, pero la segunda cambia completamente. Podemos apreciar en las imágenes dos contenedores con dos data flows diferentes que se explican en las siguientes líneas.

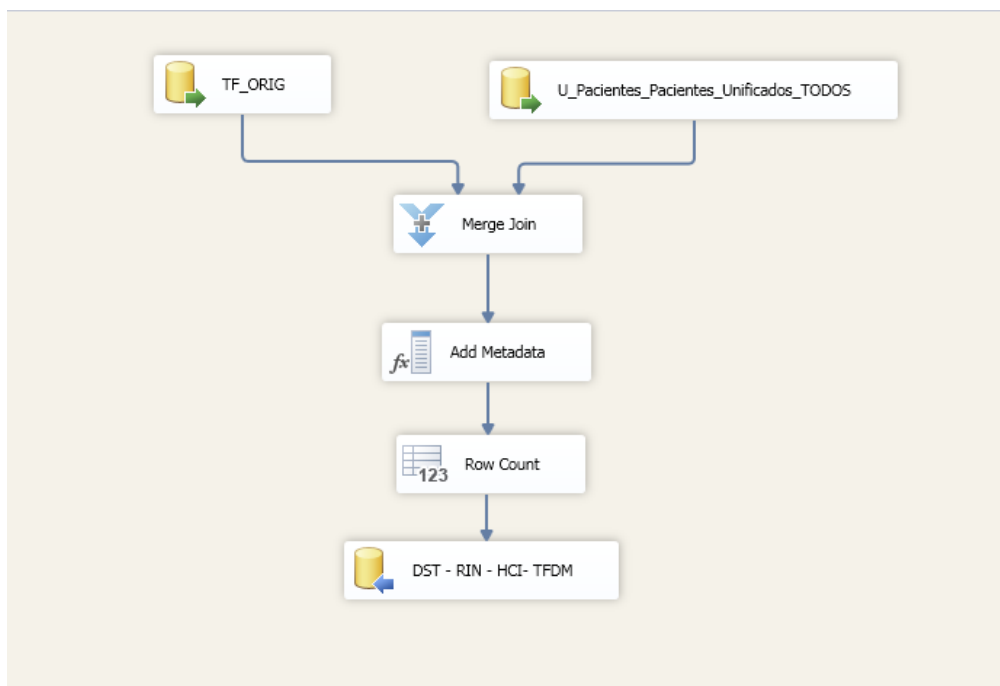


Ilustración 39 Data Flow TFDM del proceso de paso al Data Mart de una tabla de hechos

En este primer data flow, mostrado en la ilustración 39, es de una tabla de hechos, lo que hacemos es crear una tabla intermedia denominada con el prefijo TFDM_. Esta tabla contiene los datos de la TF origen, además de un campo nuevo que en este caso será el Id_Paciente_Padre, ya que cruzamos con la tabla de pacientes unificados.

Este paso es muy característico de este DM, puesto que tenemos identificadores de pacientes muy diversos según el producto fuente del que procedan. Hemos fijado unas normas de prioridad en las que se selecciona un identificador padre por encima de otro.

Si por ejemplo llega el identificador de un paciente: PLR#1234560000 por el nombre sabemos que este identificador de paciente proviene de Leire. Si en la tabla de pacientes unificados, existe una fila en la que ese identificador se relaciona con otro identificador prioritario, por ejemplo PLK#6543210000 de Lakora, entonces este identificador se guardará como Id_Paciente_Padre.

De forma que en la tabla TFDM tenemos los campos de la TF, además de campos “padre” de las dimensiones de pacientes y profesionales, que son unas dimensiones un poco especiales, como se ha podido observar a lo largo del desarrollo del trabajo.

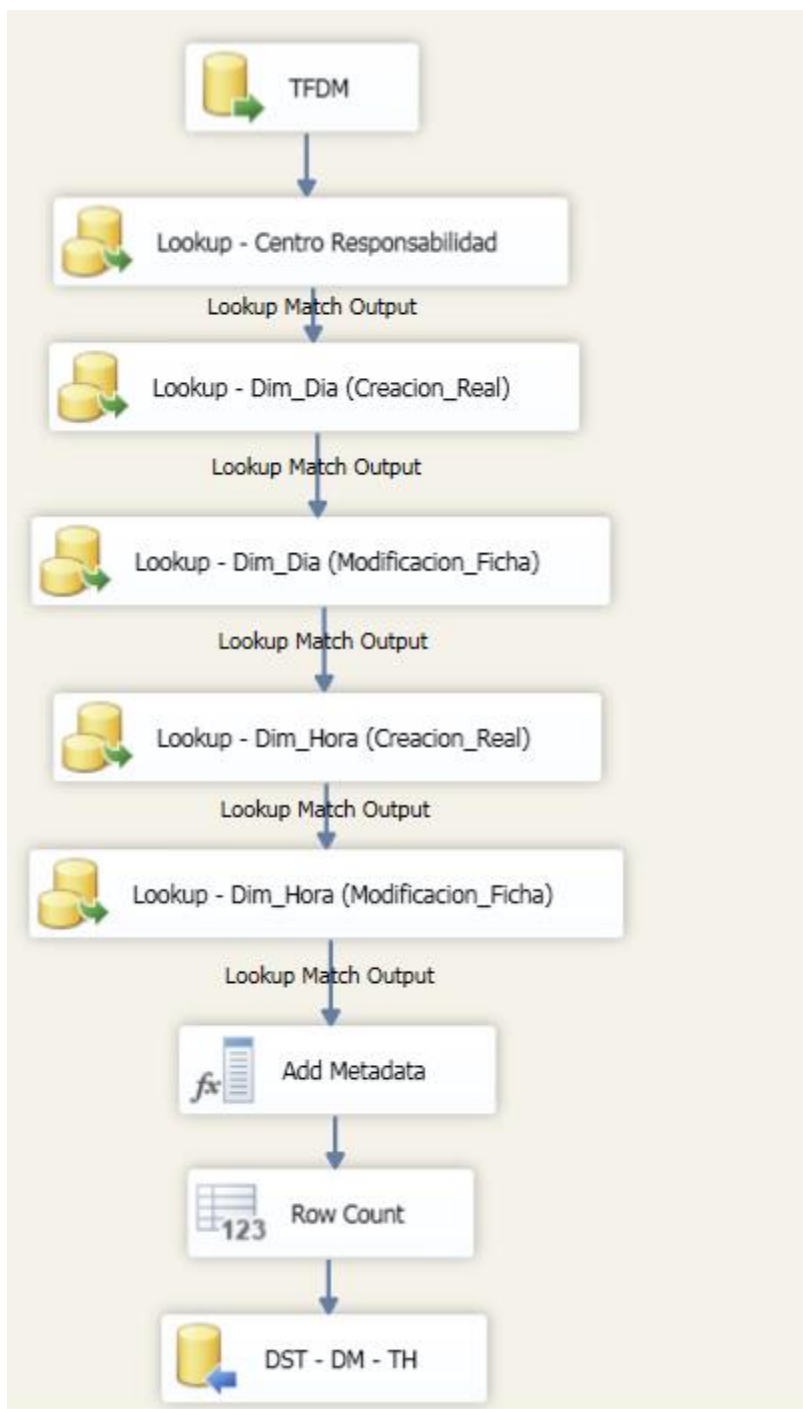


Ilustración 40 Data Flow TH del proceso de paso al Data Mart de una tabla de hechos

Este es el último y más importante flujo de datos para obtener la tabla de hechos resultante. Se realiza un lookup con cada una de las dimensiones de los que hay campos en la tabla TFDM, para que la tabla de hechos únicamente tenga campos de tipo entero. Los valores de estos campos se corresponden con los identificadores de cada dimensión.

Este ejemplo de la ilustración 40 se corresponde con la tabla TH_Evolución_Social, cruzamos con las dimensiones de fechas, día y hora. La tabla original, disponía de dos campos que expresaban una fecha, la fecha de creación y la fecha de modificación de la ficha. Para cada una de estas fechas disponíamos de dos campos, uno de tipo date y otro de tipo time. Mientras que en la nueva tabla dispondremos de dos valores enteros uno para el día y otro

para la hora, que facilitarán todos los cálculos que queramos realizar con estos campos de la tabla de hechos. Por ejemplo si queremos extraer solo las filas de la tabla cuya creación sea posterior a 2016, bastará con poner una condición sin necesidad de realizar *converts* ni otro tipo de tratamientos. Además la dimensión de día, dispone de varios campos pre calculados como lo son, un día antes, un día después, una semana antes, una semana después, el último viernes... etc. Campos que son muy útiles a la hora de utilizar el Data Mart.

6.2.3 Bases de Datos de configuración y estado del modelo BI

En el apartado anterior, en los templates de Extracción y Validación se hace referencia a las bases de datos de configuración y *logs* del sistema de BI. En este apartado se explicarán en detalle cada una de ellas y que procedimientos almacenados tienen para que los templates funcionen correctamente y puedan ser reutilizables, realizando apenas dos cambios.

6.2.3.1 RIN_conf

En esta base de datos, ilustración 41, se almacena la configuración con la que se ejecutan los procesos de Visual Studio. Está formada por varias tablas, que permiten asociar cada proceso dtsx su configuración.

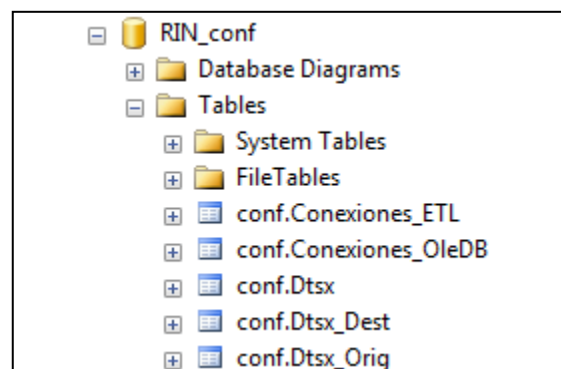


Ilustración 41 Base de datos RIN_conf

En primer lugar, observamos la tabla Dtsx, en esta tabla se almacenan los nombres de los diferentes procesos que conforman nuestro sistema de BI. En la ilustración 42 se observan varias filas del contenido de esta tabla, en este caso aparecen los nombres de procesos de extracción de Atenea.

Results		Messages
4	DM_Poblacional_Dim_Tipo_Procedencia	
5	E_Atenea_E_IDDCAP	
6	E_Atenea_E_IDDCEN	
7	E_Atenea_E_IDDESP	
8	E_Atenea_E_IDDPA2	

Ilustración 42 Datos de la tabla Dtsx

En segundo lugar, observamos la tabla Dtsx_Orig, está tabla asocia cada dtsx, a una o varias consultas de origen. Como se puede observar en la ilustración 43, aparece el nombre del proceso que carga la dimensión Carácter Prueba, y realiza una consulta. Además podemos apreciar las columnas FI y FF, donde se indican (si fuera necesario) la fecha de inicio y la fecha de fin, de la consulta.

	Cod_Dtsx	Cod_Orig	SQL_Orig	FI	FF
1	DM_Poblacional_Dim_Caracter_Prueba	Dim_Caracter_Prueba	SELECT DISTINCT [Cod_Caracter] ,[Desc_Caracter] FRO...		
2	DM_Poblacional_Dim_Estado_Prueba	Dim_Estado_Prueba	SELECT Cod_Estado,Desc_Estado FROM HCI.TF_Prueb...		
3	DM_Poblacional_Dim_Prestaciones	Dim_Prestaciones	SELECT AG.[Id_Prestacion],AG.[Nombre] ,ISNULL(AG.[A...		
4	DM_Poblacional_Dim_Tipo_Proceden...	Dim_Tipo_Procede...	SELECT DISTINCT [Cod_Tipo_Procedencia_Paciente] ...		
5	E_Atenea_E_IDDCAP	IDDCAP	SELECT REFERENCIA, NOMBRE, NOMBREABRE, COL...		
6	E_Atenea_E_IDDCEN	IDDCEN	select [ZONA],[GERENCIA],[CENTRO],[NOMBRE],[DOMI...		
7	E_Atenea_E_IDDESP	IDDESP	select [CODIGOESP],[DESCRIPCIO],[ESPINSALUD],[ESP...		
8	E_Atenea_E_IDDPA2	IDDPA2	select ZONA,GERENCIA,CENTRO,CONSULTORIO,NIF,A...		

Ilustración 43 Datos de la tabla Dtsx_Orig

En tercer lugar, observamos la tabla Dtsx_Dest. Esta tabla relaciona cada proceso, con una o varias tablas destino. En esta tabla destino, será donde se almacenará el resultado del proceso. Vemos algunos datos de esta tabla en la ilustración 44.

	Cod_Dtsx	Cod_Dest	Tabla_Dest
1	DM_Poblacional_Dim_Caracter_Prueba	Dim_Caracter_Prueba	HCI.Dim_Caracter_Prueba
2	DM_Poblacional_Dim_Estado_Prueba	Dim_Estado_Prueba	HCI.Dim_Estado_Prueba
3	DM_Poblacional_Dim_Prestaciones	Dim_Prestaciones	HCI.Dim_Prestaciones
4	DM_Poblacional_Dim_Tipo_Procedencia	Dim_Tipo_Procedencia	HCI.Dim_Tipo_procedencia
5	E_Atenea_E_IDDCAP	IDDCAP	atenea.E_IDDCAP
6	E_Atenea_E_IDDCEN	IDDCEN	atenea.E_IDDCEN
7	E_Atenea_E_IDDESP	IDDESP	atenea.E_IDDESP
8	E_Atenea_E_IDDPA2	IDDPA2	atenea.E_IDDPA2

Ilustración 44 Datos de la tabla Dtsx_Dest

En cuarto y último lugar, tenemos la tabla Conexiones_OleDB, se muestra un ejemplo de su contenido en la ilustración 45. Esta tabla solo se emplea en casos especiales. Cuando tenemos una réplica, sobre varias bases de datos, es decir, que nos proporcionan la información segmentada en varias bases de datos. Estas bases de datos siguen la misma estructura y los mismos nombres de tablas, de modo que podemos reutilizar la misma extracción para todas estas tablas y de hecho, introduciremos todos los datos en una sola tabla destino. En estos casos, emplearemos la tabla Conexiones_OleDB para cargar las conexiones a cada una de estas bases de datos. Solamente se emplea en réplicas de HCI y Atenea.

	Entorno	Producto	Servidor	Id_Base_Datos	Base_Datos
1	Desarrollo	Atenea	CENTSQLDES08,51433	19001	OMIAPW19001_EXPLOTTBI
2	Desarrollo	Atenea	CENTSQLDES08,51433	25009	OMIAPW25009_EXPLOTTBI
3	Desarrollo	HCI	CENTSQLDES08,51433	1	HistoriasClinicas01_EXPL...
4	Desarrollo	HCI	CENTSQLDES08,51433	3	HistoriasClinicas03_EXPL...
5	Desarrollo	Atenea	CENTSQLDES08,51433	25055	OMIAPW25055_EXPLOTTBI
6	Desarrollo	Atenea	CENTSQLDES08,51433	70001	OMIAPW70001_EXPLOTTBI
7	PreProd...	Atenea	CENTSQLDES08	19001	OMIAPW19001_EXPLOTTBI
8	PreProd...	Atenea	CENTSQLDES08	25009	OMIAPW25009_EXPLOTTBI

Ilustración 45 Datos tabla Conexiones_OleDB

Una vez hemos observado la estructura de las bases de datos, que almacenan la configuración de cada proceso, vamos a observar cómo se utilizan.

La ilustración 46, representa un contenedor del *template* de la ilustración 20, empleado para la fase de Extracción. En este contenedor se cargan las variables Tabla1_DEST, SQL1_ORIG y Conexiones_ETL.

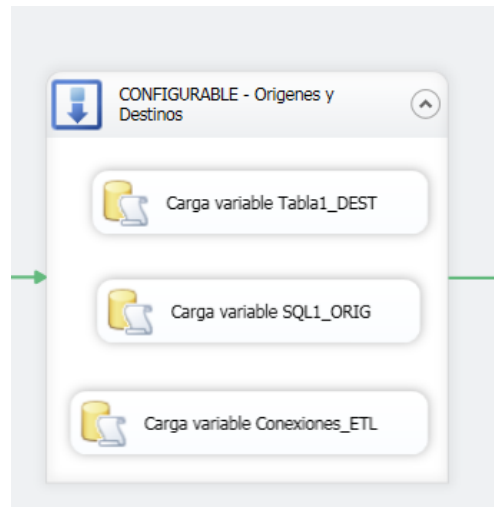


Ilustración 46 Contenedor configurable (Orígenes y Destinos) del template de Extracción

Para la carga de estas variables, emplearemos los procedimientos almacenados de la base de datos RIN_conf. Adjunto a la derecha una imagen con todos ellos.

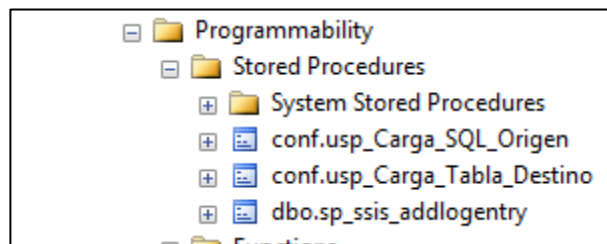


Ilustración 47 Base de datos RIN_conf. Procedimientos almacenados

Volviendo al contenedor del template, en la primera caja, un Execute SQL Task, se ejecuta el procedimiento almacenado usp_Carga_SQL_Origen. Dicho procedimiento devuelve el valor de la tabla destino y lo introducimos en una variable local de Visual Studio. Esta variable será la que posteriormente empleemos para el Destination Assistant.

En la segunda caja también un Execute SQL Task, se ejecuta el procedimiento usp_Carga_Tabla_Destino. Dicho procedimiento devuelve la consulta que realizamos sobre la tabla origen y lo introducimos en una variable local de Visual Studio. Esta variable será la que posteriormente empleemos para el Source Assistant.

En la tercera caja también un Execute SQL Task, se ejecuta el procedimiento una consulta, sobre la tabla Conexiones_OleDb. Dicha consulta devuelve una tabla con varias conexiones a bases de datos que introducimos en una variable local de Visual Studio. Esta variable será la que posteriormente empleemos para realizar el bucle, dentro del cual está el data flow. Utilizamos este bucle, para realizar el mismo proceso de extracción sobre tablas con la misma

estructura en diferentes bases de datos. (Esto sucede en las réplicas de Atenea y HCI, como se ha explicado anteriormente.) Todas estas tablas de origen con la misma estructura, se introducen en una sola tabla, que es la tabla destino. En la ilustración 48, se puede observar el bucle que itera sobre las diferentes bases de datos extraídas de la tabla Conexiones_OleDB.

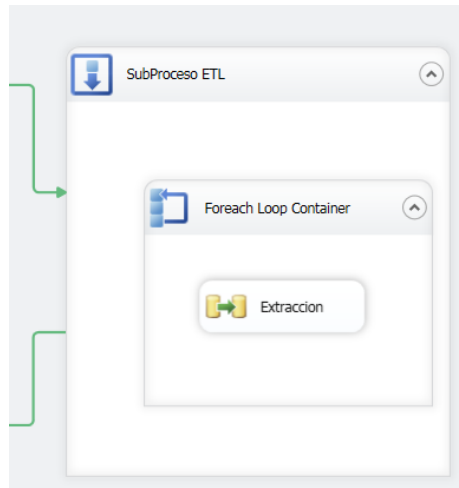


Ilustración 48 Bucle que itera sobre las bases de datos del template de Extracción

6.2.3.2 RIN_log

La base de datos RIN_log se muestra en la ilustración 49 y como su propio nombre indica, almacena los estados de ejecución de los procesos dtsx. Para cada extracción y validación, se insertan en las tablas Estadísticas y Estado_Ejecucion datos sobre su funcionamiento. Tanto si se han ejecutado correctamente, como si no lo han hecho, y en este caso se especifica el motivo del fallo y donde se ha producido.

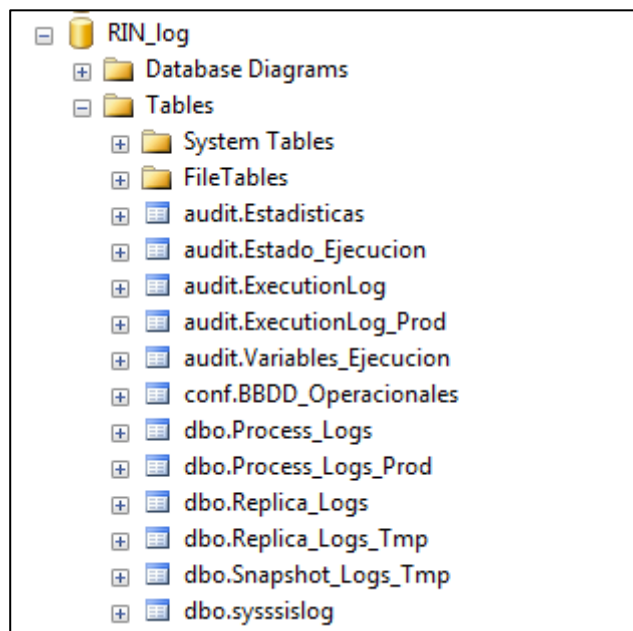


Ilustración 49 Base de datos RIN_log

La tabla Estadísticas, almacena datos sobre el número de registros que tiene la tabla de origen y el de la tabla destino. Además de la resta de ambos. (Si en el data flow hay más de una tabla de origen, o de destino también se cuentan.) En los datos que se muestran a continuación en la ilustración 50, podemos observar las últimas cuatro filas seleccionadas, estas filas se corresponden con el flujo de datos del proceso de extracción de la tabla datos generales paciente. La primera fila significa, cuya Entrada_Salida vale 4, introduce el número de filas de la tabla origen y el tiempo que se ha tardado en obtener los datos. La segunda Entrada_Salida igual a 0, significa que son datos de entrada, en este caso el total de filas de la tabla origen. La tercera fila, Entrada_Salida igual a 1, significa que son datos de salida, en este caso el número de filas que cargamos en la tabla destino. Por último, la fila cuya Entrada_Salida es 3, es la resta de las filas de origen menos las filas de destino.

	LogID	Entrada_Salida	Num_Registros	dtsc	Descripcion	Tiempo_Milseg	Min_Filas_Por_Seg	Mean_Filas_Por_Seg	Max_Filas_Por_Seg	Fecha_Carga
52	363	4	800976	E_HCI_E_F_Exploraciones_Cabece	Extraccion	25933	12928	30886	81260	2017-05-24 07:48:52
53	367	0	1310	E_HCI_E_mFichas	Almacena Num_Registros_ORIG01	0	NULL	NULL	NULL	2017-05-24 07:54:07
54	367	1	1310	E_HCI_E_mFichas	Almacena Num_Registros_DEST01	0	NULL	NULL	NULL	2017-05-24 07:54:07
55	367	3	0	E_HCI_E_mFichas	Calcula diferencia Num_Registros	0	NULL	NULL	NULL	2017-05-24 07:54:07
56	368	0	3562	E_HCI_E_mPrestaciones	Almacena Num_Registros_ORIG01	0	NULL	NULL	NULL	2017-05-24 07:55:08
57	368	1	3562	E_HCI_E_mPrestaciones	Almacena Num_Registros_DEST01	0	NULL	NULL	NULL	2017-05-24 07:55:08
58	368	3	0	E_HCI_E_mPrestaciones	Calcula diferencia Num_Registros	0	NULL	NULL	NULL	2017-05-24 07:55:08
59	369	0	453	E_HCI_E_mSecciones	Almacena Num_Registros_ORIG01	0	NULL	NULL	NULL	2017-05-24 07:55:41
60	369	1	453	E_HCI_E_mSecciones	Almacena Num_Registros_DEST01	0	NULL	NULL	NULL	2017-05-24 07:55:41
61	369	3	0	E_HCI_E_mSecciones	Calcula diferencia Num_Registros	0	NULL	NULL	NULL	2017-05-24 07:55:41
62	376	4	2827	E_HCI_E_Datos_Generales_Pacien	Extraccion	31	90473	91193	90473	2017-05-24 08:00:25
63	376	0	2827	E_HCI_E_Datos_Generales_Pacien	Almacena Num_Registros_ORIG01	0	NULL	NULL	NULL	2017-05-24 08:00:41
64	376	1	2827	E_HCI_E_Datos_Generales_Pacien	Almacena Num_Registros_DEST01	0	NULL	NULL	NULL	2017-05-24 08:00:41
65	376	3	0	E_HCI_E_Datos_Generales_Pacien	Calcula diferencia Num_Registros	0	NULL	NULL	NULL	2017-05-24 08:00:41

Ilustración 50 Datos tabla Estadísticas

La tabla Estado_Ejecucion, almacena datos sobre el número de registros que tiene la tabla de origen y el de la tabla destino. El campo más importante de esta tabla es el campo Estado. Dicho campo indica el estado de la ejecución: 0 si el proceso está ejecutándose, 1 si el proceso se ha ejecutado con éxito, 2 si el proceso ha sufrido errores (va acompañado del campo Task_Fallo que especifica donde se ha producido el error). Vemos algunos datos de ejemplo en la ilustración 51.

	LogID	Padre_LogID	dtsc	Estado	Task_Fallo	Usuario_Carga	Fecha_INS	Fecha_UPD
1	1	NULL	E_Leire_E_TH142	2		ADMON-CFNAVARRA\d111062	2017-05-16 11:49:39	2017-05-16 11:49:47
2	2	NULL	E_Leire_E_TH142	1	NULL	ADMON-CFNAVARRA\d111062	2017-05-16 11:53:11	2017-05-16 11:53:12
3	3	NULL	E_Leire_E_TH142	1	NULL	ADMON-CFNAVARRA\d111062	2017-05-16 11:56:46	2017-05-16 11:56:47
4	4	NULL	E_Leire_E_TH142	2	ESTATICO - Log finalizado,LOG - Cambia estado a ...	ADMON-CFNAVARRA\d111062	2017-05-16 11:59:40	2017-05-16 11:59:41
5	5	NULL	E_Leire_E_TH142	1	NULL	ADMON-CFNAVARRA\d111062	2017-05-16 12:05:53	2017-05-16 12:08:10
6	6	NULL	E_Leire_E_TH142	1	NULL	ADMON-CFNAVARRA\d111062	2017-05-16 12:10:28	2017-05-16 12:10:29
7	7	NULL	E_Atenea_E_IDDCAP	0	NULL	ADMON-CFNAVARRA\d111062	2017-05-17 07:38:53	NULL
8	8	NULL	E_Atenea_E_IDDCAP	1	NULL	ADMON-CFNAVARRA\d111062	2017-05-17 07:44:18	2017-05-17 07:44:20
9	9	NULL	E_Atenea_E_IDDCAP	2		ADMON-CFNAVARRA\d111062	2017-05-17 07:50:18	2017-05-17 07:50:20
10	10	NULL	E_Atenea_E_IDDCAP	2		ADMON-CFNAVARRA\d111062	2017-05-17 07:51:37	2017-05-17 07:51:39
11	11	NULL	E_Atenea_E_IDDCAP	2		ADMON-CFNAVARRA\d111062	2017-05-17 07:52:33	2017-05-17 07:52:35
12	12	NULL	E_Atenea_E_IDDCAP	2	ESTATICO - Log finalizado,LOG - Cambia estado a ...	ADMON-CFNAVARRA\d111062	2017-05-17 07:52:59	2017-05-17 07:53:01
13	13	NULL	E_Atenea_E_IDDCAP	2	ESTATICO - Log finalizado/LOG - Cambia estado a ...	ADMON-CFNAVARRA\d111062	2017-05-17 07:53:51	2017-05-17 07:53:53
14	14	NULL	E_Atenea_E_IDDCAP	2	/ ESTATICO - Log finalizado / LOG - Cambia estad...	ADMON-CFNAVARRA\d111062	2017-05-17 07:54:23	2017-05-17 07:54:25
15	15	NULL	E_Atenea_E_IDDCAP	2	ESTATICO - Log finalizado / LOG - Cambia estado ...	ADMON-CFNAVARRA\d111062	2017-05-17 07:54:53	2017-05-17 07:54:54

Ilustración 51 Datos tabla Estado_Ejecucion

Las tablas recientemente explicadas, se rellenan en los pasos del template de los procesos de Extracción y Validación que explicaré a continuación. Pero antes introduciré los procedimientos almacenados de la base de datos RIN_log, se observan en la ilustración 52, que empleamos en los siguientes pasos.

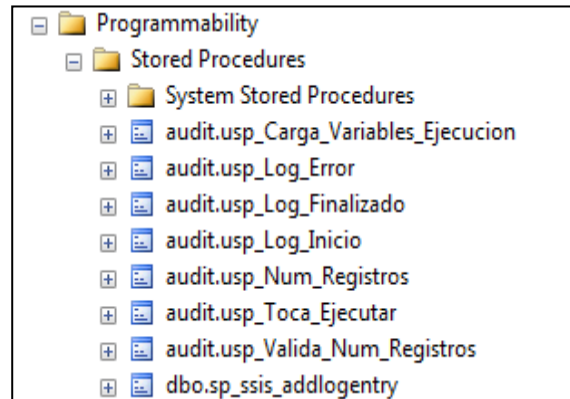


Ilustración 52 Base de datos RIN_log

A continuación se muestra la ilustración 53 con el primer y último contenedor del template de Extracción y Validación. En ambos contenedores tenemos la caja con el prefijo LOG-. Dicha caja, lo que hace es ejecutar un procedimiento almacenado, para introducir datos en la tabla Estado_Ejecucion. La primera de estas dos cajas, llama al procedimiento usp_Log_Inicio, que inserta una nueva línea en esta tabla, con el estado a 0. En el último paso del template, en la caja del segundo contenedor de la imagen, cuando ya se han introducido los tratados datos en la tabla destino, se llama al procedimiento usp_Log_Finalizado, que actualiza la fila recientemente introducida con el estado a 1.

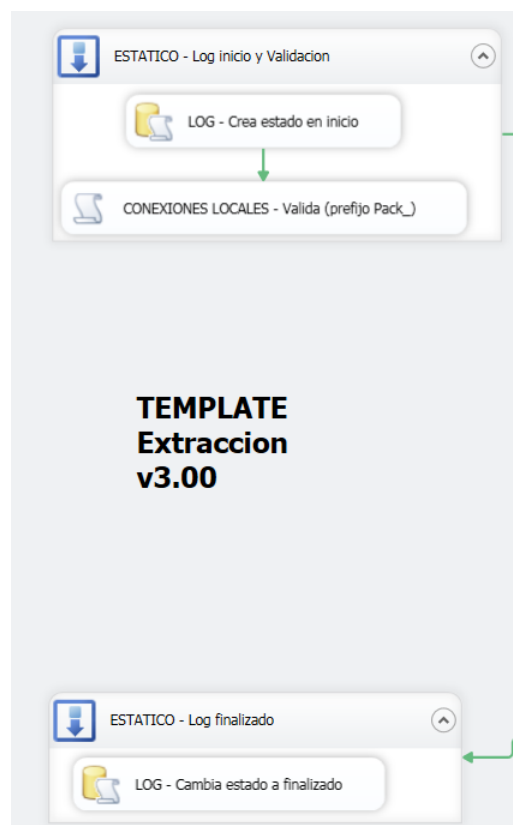


Ilustración 53 Contenedor inicial y final del template de Extracción

Además si durante la ejecución del proceso, hay algún error, se llama a la caja preparada para gestionar errores. Dicha caja se muestra a continuación en la ilustración 54, se crea dentro del Event Handlers y lo que hace es llamar al procedimiento usp_Log_Error que actualiza el estado a 2, añadiendo en el campo Task_Error el lugar (caja y contenedor) donde se ha producido el error.

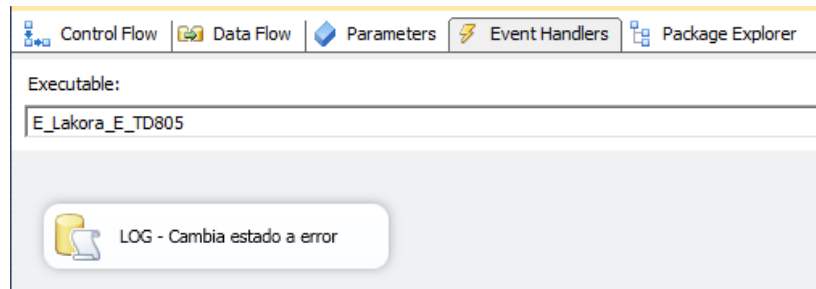


Ilustración 54 Event Handlers del template de Extracción

En cuanto a la tabla de Estadísticas, se rellena en un contenedor posterior al data flow, pero anterior al de finalización que he explicado anteriormente. En este contenedor, mostrado en la imagen 55, básicamente introducimos en la tabla Estadísticas los datos recabados a lo largo de la ejecución del proceso. Tales como el número de filas que extraemos de la tabla origen y el número de filas que introducimos en la tabla destino. Además introducimos en la tabla el dato resultante la diferencia entre este número de filas. Para realizar estos inserts sobre la tabla, se llama al procedimiento usp_Num_Registros. Como podemos observar en el contenedor, tenemos una caja más que se utiliza en el caso de que empleemos tablas intermedias durante el data flow. El número de filas de estas tablas también se introducirá, y afectará al cálculo de la diferencia, pues lo que nos interesa es el cómputo total de filas que se cargan y las que llegan al final. Para realiza dicho calculo, se emplea el procedimiento usp_Valida_Num_Registros.

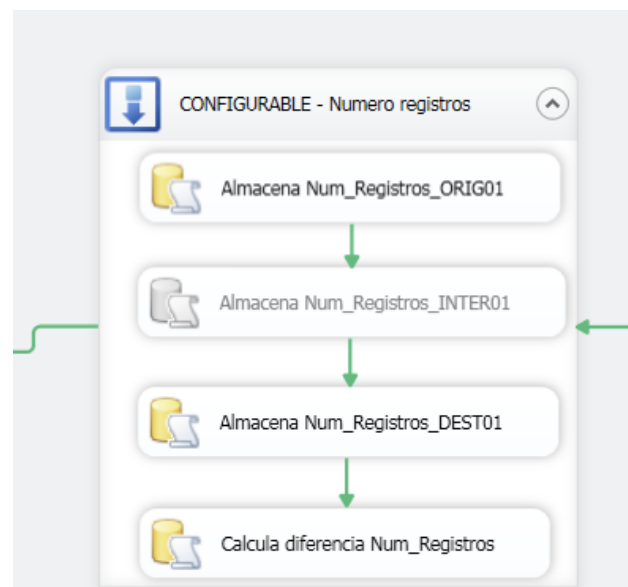


Ilustración 55 Contenedor penúltimo del template de Extracción

7. Funcionamiento

Una vez explicado cómo se ha diseñado y realizado el Data Mart. A continuación explicaré algunas de sus utilidades. Como ya he dicho el propio Data Mart es en sí un producto que el cliente (médicos, jefes de plantas, jefes de hospitales...) utiliza. Pero además nosotros les facilitamos información a través varios informes. Por ejemplo el siguiente informe elaborado con la herramienta Tableau, se muestra a médicos con pacientes crónicos.

Este cuadro de mandos de pacientes crónicos, muestra varios indicadores como lo es Nº de pacientes diana, que sería el número total de pacientes. Este indicador se calcula para cada CIAS (Código de identificación de asistencia sanitaria) correspondiente a una plaza de un médico. Cada médico solo tiene acceso a los datos de sus pacientes, arriba a la derecha se indica el CIAS a los que corresponden los datos. A su vez cada CIAS pertenece a una zona, que también se indica arriba a la derecha, junto con la fecha del cálculo de los datos.

Pero si un médico solamente pudiera ver sus datos, no tendría con que compararlos, salvo quizás con sus datos antiguos. Para ellos resulta muy interesante además de ver sus resultados, poder ver los de su zona y a mayor escala los de toda Navarra, de modo que esta información también se les proporciona en las columnas Zona y Navarra que aparecen en el cuadro. La columna de UBA sería la correspondiente al CIAS del médico.

La generación de un cuadro de mandos, o de un panel de control supone varias fases. La primera de ellas es definir los indicadores, es el cliente quien define los indicadores y nos los solicita en las reuniones que realizamos con ellos al inicio de cada *Sprint*. A continuación, se realiza el cálculo del indicador encomendado por el cliente. Para cada indicador generamos un procedimiento que se almacena en la base de datos DM_Poblacional_TH. Este procedimiento ataca a las tablas de hechos y a las dimensiones del Data Mart y aplicando las condiciones que ha especificado el cliente, se obtiene el resultado. El siguiente paso es crear el cuadro de mandos en la herramienta de Tableau. Esta herramienta obtiene los datos de una tabla y los muestra tal y como se lo indiquemos. Para que desde Tableau podamos obtener los datos de todos los indicadores, creamos una tabla en el DM_Poblacional_TH que guarda todos estos datos. Cada uno de los procedimientos que se han mencionado anteriormente, actualiza la tabla realizando *inserts* de los nuevos valores calculados. Esta tabla almacenará los resultados de un mismo indicador para diferentes fechas, para que el cuadro de mandos tenga la opción de ver resultados pasados.



CUADRO DE MANDOS DE CRÓNICOS

CIAS: -
ZONA: -

Fecha
2017/05/28

Identificación Clasificación	UBA	Zona	Navarra	Formación en cuidados	UBA	Zona	Navarra
Nº Pacientes Diana	-	-	-	% Formados en autocuidados	-	-	-
Nº Pacientes Cubo azul	-	-	-	% persona cuidadora con cuidados eficaces	-	-	-
Nº Pacientes Cubo rojo	-	-	-	Nuevos modelos de atención	UBA	Zona	Navarra
% Pacientes Cubo azul	-	-	-	% contactados por AP en 24 h. tras alta hosp.	-	-	-
% Pacientes Cubo rojo	-	-	-	Efectividad/resultados	UBA	Zona	Navarra
% con valoración global	-	-	-	Tasa de Urgencias en AE	-	-	-
% consta Nivel de Severidad Clínica	-	-	-	Tasa de un ingreso hospitalario	-	-	-
% con dtco necesidades cuidados sanitarios	-	-	-	Media de Dias de estancias por paciente	-	-	-
% con dtco necesidades sociales	-	-	-	Seguridad del paciente	UBA	Zona	Navarra
% clasificados como severos/paliativos	-	-	-	% de pacientes Polimedicados	-	-	-
% con Especialista de referencia	-	-	-	Control Clínico	UBA	Zona	Navarra
% con Especialista de Med Interna	-	-	-	% Pacientes con demencia con GDS	-	-	-
% con Especialista de geriatría	-	-	-	% Pacientes con Insuf. cardiaca con NYHA	-	-	-
% con Especialista de neurología	-	-	-				
% con Especialista de cardiología	-	-	-				
% con Especialista de endocrinología	-	-	-				
Diagnostico y tratamiento integral	UBA	Zona	Navarra				
% con plan de intervencion integral	-	-	-				
% con Plan de Cuidados	-	-	-				
% con Evaluación Capacidad Funcional	-	-	-				
% severos con Info sobre Persona Cuidadora	-	-	-				
% persona cuidadora con sobrecarga valorada	-	-	-				

Ilustración 56 Cuadro de Mandos Crónicos

Un ejemplo de un cuadro de mandos, es el de la imagen superior que realiza cálculos de indicadores para pacientes crónicos. Podemos apreciar que existen varios tipos de indicadores (franja azul oscura) por ejemplo indicadores de *identificación clasificación* o *formación en autocuidados*. Además, para cada indicador, tenemos un texto representativo por ejemplo *nº de pacientes diana*, y tres resultados (los tres guiones). El primero, el UBA, se calcula para los pacientes de la plaza del médico que introduzcamos (arriba a la derecha aparece el CIAS). El segundo resultado, por zona, se calcula para los pacientes de la zona básica a la que pertenece el CIAS en cuestión (se indica esta zona arriba a la derecha). En tercer lugar, tenemos el cálculo para todos los pacientes de Navarra. Además el cuadro de mandos de pacientes crónicos permite seleccionar la fecha para la cual queremos calcular todos los indicadores (sobre unas fechas ya establecidas que los realizan cada mes).

Por último, este cuadro de mandos se utiliza a diario en el Servicio Navarro de Salud y como se puede apreciar en la imagen dispone una visualización sencilla de un conjunto de datos muy valiosos. Además ofrece privacidad, en cuanto a que un médico no puede ver los resultados de un CIAS que no le corresponda. Pero a la vez permite comparaciones, mostrando los cálculos de los mismos indicadores para grupos de pacientes más generales, además de para otras fechas. Esto ayuda a cada médico y a los encargados de cada unidad, a medir el rendimiento de su sección, pudiendo ver sus puntos fuertes y sus puntos débiles respecto a su zona o a Navarra, para poder reforzar los segundos.

Adjunto en el anexo 10.3, el cálculo del indicador que calcula el *porcentaje de pacientes polimedicados*. Un ejemplo del paso en el que generamos el procedimiento almacenado, en el proceso explicado anteriormente.

8. Conclusiones y líneas futuras

Como se puede ver a lo largo del documento desarrollar un modelo de BI es complicado. Existen modelos teóricos como el de Kimball que se pueden tomar como referencia, pero estos modelos nunca tienen en cuenta todas las posibles casuísticas con las que nos podemos encontrar al emplearlos. En estos casos es cuando tenemos que recurrir a nuestra propia experiencia y creatividad para adaptar estos modelos a cada caso. En nuestro modelo de BI, probablemente uno de los más complejos, ya que el ámbito sanitario es más exigente que otros. Por ejemplo, la precisión de los datos es muy estricta, no se pueden confundir los datos de dos pacientes, pues puede ser un error mortal. En nuestro caso no se emplean los datos del sistema de BI, para obtener información crítica de un paciente, si no que esto se realiza del producto concreto del que dependa el médico en cuestión, pero en un futuro sería lo ideal que todos los productos se integraran y obtuvieran su información del SE. Otro dato propio de salud es que se tienen en cuenta fuentes muy diversas y muy antiguas, de hecho las primeras bases de datos que existen se dedicaban al mundo sanitario, de modo que su integración es más complicada que la de bases de datos de fuentes más actuales.

En cuanto a las líneas futuras de este proyecto se abren varios caminos. Por un lado existen proyectos presentes y futuros de mejora y mantenimiento de todo el sistema de explotación. Hay que continuar trabajando en la mejora del modelo de BI existente. Un punto fuerte de mejora es la calidad del dato, pues no todos los campos pasan un estricto control de calidad, si lo hacen los más importantes como los identificadores de los pacientes y de los profesionales, pero lo deberían hacer todos.

Por otro lado, se persigue la integración de los datos con los operacionales para que se retroalimenten mutuamente y obtener una precisión mayor en la obtención de datos en todos los productos. La retroalimentación mencionada, se refiere a que al comprobar un dato en el modelo de BI, si no es correcto, por ejemplo un DNI de tres dígitos, este se derive al producto del que ha sido extraído. Este producto fuente, conociendo sus fallos, podrá añadir una restricción para que en los campos donde recoge el DNI, se realice la comprobación de que el dato que se inserte es realmente un DNI.

Por otro lado, existen numerosos proyectos para explotar el modelo de BI. Se puede explotar de dos formas, directamente e indirectamente. Cuando digo directamente me refiero a atacar a los datos que obtenemos con el sistema, mientras que indirectamente me refiero a que sea el modelo el que nos sirva los datos, como por ejemplo con los informes.

También, se pretenden realizar nuevos cuadros de mando y paneles de control, lo que conllevará nuevos indicadores. Actualmente existen listas con parte de estos nuevos indicadores que le interesan al cliente, de modo que en líneas futuras se prevé que se invertirá gran cantidad de tiempo en el desarrollo de nuevos informes.

Además en la explotación directa de los datos, se están realizando pequeños proyectos de minería de datos, como por ejemplo uno para clasificar los pacientes de salud mental. Pero como en el caso de los indicadores, se tiene en cartera numerosos proyectos muy ambiciosos sobre el análisis de los datos para generar predicciones. Aunque son complicadas las predicciones en el ámbito sanitario, ya que sobre cada paciente, los mismos medicamentos,

pueden actuar de formas muy diferentes. También se espera que en un futuro, se invierta en minería de datos.

9. Bibliografía y referencias

- [1] Clasificación internacional de enfermedades/ International Statistical Classification of Diseases and Related Health Problems, <http://www.who.int/classifications/icd/en/> [accedida en junio de 2017]
- [2] World Organization of National Colleges, Academies, and Academic Associations of General Practitioners/Family Physicians, "ICPC-2-R: International Classification of Primary Care", Oxford medical publications, 2nd. Edition, Oxford University Press, 2005.
- [3] SNOMED CT, <https://confluence.ihtsdotools.org/display/DOC> [accedida en junio de 2017]
- [4] Health Level-7, <http://www.hl7.org/> [accedida en junio de 2017]
- [5] Oleg S. Pinykh, "Digital Imaging and Communications in Medicine: A Practical Introduction and Survival Guide", 1st edition, Springer Publishing Company, Incorporated, 2008.
- [6] Ralph Kimball and Margy Ross, "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling", 3rd edition, Wiley Publishing, 2013.
- [7] William H. Inmon, "Building the Data Warehouse", 4th edition, Wiley Publishing, 2005.
- [8] Herramienta Microsoft Power BI <https://powerbi.microsoft.com/es-es/>
- [9] Herramienta BI Tableau <https://www.tableau.com/>
- [10] Cuadrante mágico de Gartner <http://optimalbi.com/blog/2017/02/17/gartner-magic-quadrant-for-business-intelligence-2017-cloud-is-coming-slowly/>

10. Anexos

En este apartado se añaden los anexos a los que se ha hecho referencia a lo largo del trabajo.

10.1 Requerimientos del cliente, solicitudes

En los siguientes anexos se presentan los requerimientos del cliente. En GB trabajamos con la aplicación llamada Gesprona, como medio para que los clientes realicen sus solicitudes. Posteriormente los trabajadores imputamos las horas de trabajo sobre el identificador que se le da a cada solicitud del cliente. Los requerimientos se agrupan en varios grupos.

- Planificación general: son las solicitudes principales y en las que más tiempo se quiere que invierta el equipo
- Bolsa de horas: tareas más pequeñas que al cliente le gustaría que se realizarán pero que no son esenciales. Suelen ocupar un total de horas menor que las de planificación general.
- Tareas para el RTec (responsable técnico)
- Tareas a impulsar por el Gprod (Gestor de producto)

En estas tablas de solicitudes, se puede observar el cliente, el gestor de producto y los desarrolladores nos hemos ido adaptando la metodología, optimizando nuestro tiempo (las reuniones son mucho más breves y concisas). Pues en los primeros *sprints* las solicitudes eran menos claras y en los últimos son más precisas.

De hecho, el equipo ha sido una prueba de concepto de la implantación de metodologías ágiles y dado el éxito que ha tenido, ahora somos la referencia para otros grupos en los que se está implantando actualmente.

10.1.1 Sprint 1

ID Trabajo	Título del Trabajo	Requerimiento a completar
18576	18576 Sistema de Explotacion Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	<ul style="list-style-type: none">• Avanzar el desarrollo de la dimensión de profesionales. Se estima que se implementará hasta el paso de unificación (quedará sin completar la implementación de parte de esta dimensión)• Desarrollar la unión de la dimensión de pacientes con los hechos de

		Citas AE de versión 2.x
19056	Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	<ul style="list-style-type: none"> • Completar la petición 12482 (llevar a producto la Excel de perfiles) • Realizar el diseño de petición 9480 de una tabla de hechos de Eunate (quedará sin hacer la implementación o desarrollo propio de la tabla). En el diseño se tendrá en cuenta la petición 13084 (especialidad de referencia HCI), para dejar preparado el diseño de campos en la tabla de hechos para cuando se implemente especialista referencia HCI.

Planificación “Bolsa de Horas”

ID Trabajo	Título del Trabajo	Requerimiento a completar
19056	Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	<ul style="list-style-type: none"> • Petición 12996 Reducir tiempos de carga. Se completará la mejora de la carga de la Analítica (quedarán sin refactorizar otros puntos pesados de la carga) • Completar la Petición 13084 Modificar Tipoo dato en campos th_derivación_saludmental

Otras tareas RTec

ID Petición	Título de la petición	Se avanzará en...
12216	Incluir datos de observa en la base de datos poblacional	<ul style="list-style-type: none"> • Informarnos sobre qué es Observa, sobre su réplica, sus datos.
12481	Anonimizar las tablas de report_PdCD_Cronicos_AP y de AE	<ul style="list-style-type: none"> • Reunión con el cliente para borrar todos los permisos existentes. De facto, esto supondrá que ningún usuario verá las tablas de Report. Quedará para posterior Sprint mover las tablas Report a otra BD distinta del DataMart

Tareas a impulsar

ID Trabajo	Título del Trabajo	
17367	EV Sistema de Explotación analizar mejora Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	<ul style="list-style-type: none"> • Trabajar requerimientos funcionales: estructura funcional, CR, Ceko
17365	Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	<ul style="list-style-type: none"> • Validación de pacientes

10.1.2 Sprint 2

ID Trabajo	Título del Trabajo	Requerimiento a completar
18576	18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	<ul style="list-style-type: none"> • Añadir la dimensión de Profesionales al DM (la dimensión profesionales se termina en este sprint, con la salvedad de que utilizemos SAP como fuente de datos más adelante y habrá que hacer una pequeña adaptación) • Relacionar TH de Citas_AE con nueva Dim pacientes (es poco, es

		llevar la dim pacientes de la 3.0 a la 2.x)
17365	Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	<ul style="list-style-type: none"> Desplegar en producción (pacientes)

Tareas a impulsar

ID Trabajo	Título del Trabajo	
17367	EV Sistema de Explotación analizar mejora Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	<ul style="list-style-type: none"> Trabajar requerimientos funcionales: estructura funcional, CR, Ceco
17365	Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	<ul style="list-style-type: none"> Validación de pacientes (EqDes tiene que llenar tabla auditoría con datos)

Planificación “Bolsa de Horas”

ID Trabajo	Título del Trabajo	Requerimiento a completar
19056	Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	<ul style="list-style-type: none"> Automatizar generación perfiles por defecto [12482] Reflejar en PdC AP valor de Especialidad de Referencia de HCI [11941] Desplegar nueva versión 2.39 (parcial, solo la parte de la población Diana)
17414	17414 Evolución Estratificador Poblacional por GMA v.1.1	<ul style="list-style-type: none"> Desplegar GMA v1.1 en producción (solo va a permitir la ejecución de la

	Cambio cálculo de Severidad	segmentación)
--	-----------------------------	---------------

10.1.3 Sprint 3

Petición	Trabajo	Observaciones sobre requerimiento
13825 Disponer de todas las bases de datos anonimizadas en Desarrollo de productos de Evaluación de Resultados	19383 Disponer de todas las bases de datos anonimizadas en Desarrollo de productos de Evaluación de Resultados	<p>Avanzar. Horizonte trabajar con datos en desarrollo anonimizados, sin necesidad de ver datos de producción: mayo.</p> <p>En este Sprint se va a hacer una primera versión limitada, aunque funcional. Tendrá continuación en próximo Sprint.</p>
13865 Base de datos Aux_Ssias_BI pasa a producto SE-Poblacional	Pendiente asignar (v.2.40)	
13815 Cambio en la fórmula de cálculo del indicador Número pacientes polimedicados - CdM AP	Pendiente asignar (v.2.40)	
13704 Implementar Incorporar tabla Eunete (de atenea) en la BBDD poblacional como otra tabla	Pendiente asignar (v.2.40)	
12216 Diseño incluir datos de observa en la base de datos poblacional	Pendiente asignar (v.2.40)	Solo diseño, la implementación (petición 13946 queda fuera del alcance)
13475 Modificar el cálculo	19056 Sistema de	

del número de fármacos cos para el cdM y el PdC	Explotación Poblacional v.2.39 Generación automática de perfiles	
13318 Notificación de sistema actualizado	19056 Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	
12249 Incorporar fecha corte a tabla th_citas_ae.	19056 Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	Se amplía alcance a Todas las tablas de Hechos (a confirmar con Infraestructuras)
13947 Refactorizar solución calculo indicadores permitiendo añadir nuevos indicadores sin necesidad de despliegue	Pendiente asignar (v.2.40)	En este Sprint se diseña y se realiza una prueba de concepto, y se valorará su funcionalidad. No se implementa.

Validaciones

Trabajo	Petición
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Dimensión pacientes completa
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	Dimensión profesionales completa
18688 Evolución Sistema de Explotación Poblacional 2.38 Mejorar Auditoria y cambio interfaz con Observa	13009 Indicadores. Contabilizar para el cálculo de media de Navarra solo las ZBS incluidas

19056 Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	13441 excluir de la consulta de numero de atc diferentes (para OBSERVA) los productos que no sean medicamentos
	13190 Unir dim pacientes de versión 3.x con Citas AE de la versión 2.x
	13084 Modificación tipo dato en campos th_derviacion_saludMental

Estimaciones

Petición	Trabajo	Indicaciones sobre requerimiento
13887 Estimación esfuerzo necesario para crear un CdM de diabetes en AP	Pendiente Crear tarea	Hacer una primera estimación, a groso modo, aunque tenga posible importante desviación.
12900 Estimar coste adaptación CdM SNS a Tableau Reader	18866 Estimar coste adaptación CdM SNS-O a Tableau Reader	No ahondar más. Entregar el dato con lo que se conozca hoy, y cerrar trabajo y petición. Se asume que pueda haber desviaciones.

Documentación

Trabajo	Documentos a completar
19056 Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	DLyF Modelo Conceptual (solo añadir indicadores)
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	DLyF Documento de pruebas Modelo Conceptual

18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	DLyF Documento de pruebas Modelo Conceptual
---	---

Despliegues

Trabajo	Despliegue a realizar
19056 Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	Preproducción Producción
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Preproducción Producción

10.1.4 Sprint 4

Desarrollos

Petición	Trabajo	Observaciones sobre requerimiento
14076 Carga rápida de las tablas de Hechos "caja blanca" en la versión 3.x	19523 Sistema de Explotación Poblacional v.3.25 Carga rápida de Hechos	Se espera un avance aproximado de un tercio del total.
12216 Diseño incluir datos de observa en la base de datos poblacional	19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	
13947 Refactorizar solución calculo indicadores	19489 Sistema de Explotación Poblacional v.2.40 Nuevo	POC con CdM AP con Tableau Diseñar una solución que permite de forma fácil la agregación (el desarrollo irá a otro

permitiendo añadir nuevos indicadores sin necesidad de despliegue	sistema de Cálculo de Indicadores	Sprint)
---	-----------------------------------	---------

Documentación

Trabajo	Documentos a completar
19056 Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	DLyF Modelo Conceptual (solo añadir indicadores)
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	DLyF Documento de pruebas Modelo Conceptual
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	DLyF Documento de pruebas Modelo Conceptual

Otras tareas que podrán realizarse durante el Sprint

Estos trabajos/tareas tienen dependencias para EqDes, por lo que aunque, se marca como objetivo avanzarlas lo máximo posible, no compromete su finalización durante el sprint.

Validaciones

Trabajo	Petición
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Dimensión pacientes completa
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de	Dimensión profesionales completa

Profesionales, Plazas AE y AP, y Estructura Funcional	
---	--

Despliegues

Trabajo	Despliegue a realizar
19056 Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	Preproducción Producción
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Preproducción Producción
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales	Preproducción Producción

10.1.5 Sprint 5

Desarrollos

Petición	Trabajo	Observaciones sobre requerimiento
14076 Carga rápida de las tablas de Hechos "caja blanca" en la versión 3.x	19523 Sistema de Explotación Poblacional v.3.25 Carga rápida de Hechos	Se espera un avance aproximado de dos tercios del total.
13947 Refactorizar solución calculo indicadores permitiendo añadir nuevos indicadores sin necesidad de despliegue	19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	Avanzar en la POC, generando 5 nuevos indicadores de los solicitados en petición #12479 CdM de crónicos: Incorporar indicadores al CdM. Revisión reunión con cliente

14419 Modificación datos del PdC de esquizofrenia (campo Historia laboral y situación civil)	19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	
11830 Revisar uso de licencias de Tableau desktop en SNS-O Revisar permisos DA e impacto fin mantenimiento Tableau 8	18297 Revisar uso de licencias Tableau Desktop	Preparación de la evolución a Tableau versión 10 (solo Desktop). Pendiente tras este Sprint, la evolución del Server.
13312 Revisar Tareas de Operación	19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	DGITIP revisa las tareas de operación publicadas y nos las enviará para validar y comprobar. Una vez validadas las TO revisar los perfiles y crear los necesarios.

Documentación

Trabajo	Documentos a completar
19056 Sistema de Explotación Poblacional v.2.39 Generación automática de perfiles	Modelo Conceptual (solo añadir indicadores)
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Documento de pruebas Modelo Conceptual
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	Documento de pruebas Modelo Conceptual

Otras tareas que podrán realizarse durante el Sprint

Estos trabajos/tareas tienen dependencias para EqDes, por lo que aunque, se marca como objetivo avanzarlas lo máximo posible, no compromete su finalización durante el sprint.

Validaciones

Trabajo	Petición
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Dimensión pacientes completa
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	Dimensión profesionales completa
	Petición #14254 Revisar si todos los indicadores e interfaces apuntan a la nueva tabla de hechos de crónicos. Se envió al cliente las queries con el código de todos los indicadores para su validación

Despliegues

Trabajo	Despliegue a realizar
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Preproducción Producción
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales	Preproducción Producción

19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores:

- No se despliega en este Sprint, tras incluir modificaciones PdC Esquizofrenia y otras, se planteará desplegar a fin de Sprint 05.

10.1.6 Sprint 6

Desarrollos

Petición	Trabajo	Observaciones sobre requerimiento
14076 Carga rápida de las tablas de Hechos "caja blanca" en la versión 3.x	19523 Sistema de Explotación Poblacional v.3.25 Carga rápida de Hechos	Se espera finalizar el trabajo y dejarlo listo para despliegue y validación.
13947 Refactorizar solución calculo indicadores permitiendo añadir nuevos indicadores sin necesidad de despliegue	19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	Terminar ajustes Herramienta de indicadores Aplicar "enganche" con PdC de Esquizofrenia. (Revisión reunión con cliente)
11830 Revisar uso de licencias de Tableau desktop en SNS-O Revisar permisos DA e impacto fin mantenimiento Tableau 8	18297 Revisar uso de licencias Tableau Desktop	Entregar datos al cliente Si todo correcto, y según se determine en reunión "Planificación Implantación TABLEAU (trabajo:19913)" plantear abrir tarea para despliegue versión 10 (desktop).
13312 Revisar Tareas de Operación	19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	DGITIP revisa las tareas de operación publicadas y nos las enviará para validar y comprobar. Una vez validadas las TO revisar los perfiles y crear los necesarios.
14779 Cambiar indicador de contactados en 24 horas – añadir fecha cubo rojo	19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	Realizar cambio.

Documentación

Trabajo	Documentos a completar
19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	Entrega primera versión. Avanzar en completar Modelo Conceptual
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Entrega primera versión. Avanzar en completar Modelo Conceptual y Documento de pruebas
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	Entrega primera versión. Avanzar en completar Modelo Conceptual y Documento de pruebas

Otras tareas que podrán realizarse durante el Sprint

Estos trabajos/tareas tienen dependencias para EqDes, por lo que aunque, se marca como objetivo avanzarlas lo máximo posible, no compromete su finalización durante el sprint.

Validaciones

Trabajo	Petición
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Dimensión pacientes completa
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales, Plazas AE y AP, y Estructura Funcional	Dimensión profesionales completa
	Petición #14254 Revisar si todos los indicadores e interfaces apuntan a la nueva tabla de hechos de crónicos.

	Se envió al cliente las queries con el código de todos los indicadores para su validación
--	---

Despliegues

Trabajo	Despliegue a realizar
17365 Evolución Sistema de Explotación Poblacional v.3.0 Mejora dimensión de Pacientes	Finalizar Preproducción Producción
18576 Sistema de Explotación Poblacional v.3.1 Dimensión de Profesionales	Finalizar Preproducción Producción
19489 Sistema de Explotación Poblacional v.2.40 Nuevo sistema de Cálculo de Indicadores	Preproducción Producción

10.2 Planificación de cada sprint

Los próximos anexos, muestran detalladamente la planificación resultante de la reunión de planificación. Algunas tareas al finalizar la reunión ya están asignadas a un miembro del equipo, aunque en las tablas no aparezcan. Las filas que aparecen en rojo, son aquellas que no entran dentro de la planificación del sprint. Mientras que las que aparecen en naranja puede ser o porque no entran completas en el sprint, o porque si lo hicieron pero no se llegaron a finalizar de modo que se continuarán en el siguiente sprint.

10.2.1 Sprint 1

SPRINT 1 03/02/2017 - 20/02/2017

Id Tarea	Horas		Descripción	Aclaraciones
18.576	375 h	<input checked="" type="checkbox"/>	Construir la dimensión de profesionales	Extracción, Validación, Maestros, Unificación
12.324	8 h	<input checked="" type="checkbox"/>	Tabla precedencias	
13.658	10h	<input checked="" type="checkbox"/>	Rango fechas en bases de datos	
13.655	24h	<input checked="" type="checkbox"/>	Data sources USP	
13.666	20h	<input checked="" type="checkbox"/>	Añadir índices	
12.482	40 h	<input checked="" type="checkbox"/>	Trabajar Excel recibida por parte de negocio	
9.480	80-100 h	x	Tabla de hechos Eunat	
12.985	44 h	<input checked="" type="checkbox"/>	Anonimización CIPNA, GHA	Además de investigar funcionamiento GMA
10.078	40 h	<input checked="" type="checkbox"/>	Nueva versión de GMA (utilizar fuentes del servicio catalán)	
12.996	32 h	<input checked="" type="checkbox"/>	Reducir tiempo de ejecución	
12.216		<input checked="" type="checkbox"/>	Traer observa a BBDD de poblacional	
12.481		<input checked="" type="checkbox"/>	Anonimizar Pdc y CdM	
13.085		<input checked="" type="checkbox"/>	Renombrar un campo	
13.084	16 h	<input checked="" type="checkbox"/>	Cambiar tipo	
9.480	17 h	<input checked="" type="checkbox"/>	Diseñar	

10.2.1 Sprint 2

SPRINT 2 23/02/2017 - 13/03/2017			
Id	Horas	Descripción	Aclaraciones

Tarea				
17367	313 h	<input checked="" type="checkbox"/>	Acabar dim. Profesionales	(incluidas auditoria y documentación) dejar solo dim_pacientes_ext y quitar el campo sk_pacientes_ext (modificar maestras)
17365	45 h	<input checked="" type="checkbox"/>	Desplegar 3.0	
17415	24 h	<input checked="" type="checkbox"/>	Desplegar Excel de negocio 2.39	
17414	16 h	<input checked="" type="checkbox"/>	Subir GHA (sin cambios) versión 1.1	
13315	16 h	<input checked="" type="checkbox"/>	Vista para quitar tablas dim_pacientes	
13314	8 h	<input checked="" type="checkbox"/>	Incluir campos de GMA en dim_pacientes	
13313	32 h	<input checked="" type="checkbox"/>	Cambios GMA, anonimización	
13312	24 h	<input checked="" type="checkbox"/>	Revisar tareas de operación (perfiles)	
12216		x	Incluir datos observa en BBDD poblacional	RTEC informarse y preparar para afrontar en el siguiente sprint
11941	32 h	<input checked="" type="checkbox"/>	Incorporar segunda especialidad por paciente	Afecta a 12 procedimientos del CdM, a 1 del PdC y a las dos tablas
13475	40 h	<input checked="" type="checkbox"/>	Indicador número de medicamentos	Cambian condiciones, no tener en cuenta medicamentos como pañales o absorbentes
13477	40 h	<input checked="" type="checkbox"/>	maestro especialidades	

10.2.2 Sprint 3

SPRINT 3 16/03/2017 - 04/04/2017

Id Tarea	Horas	Descripción	Aclaraciones
17.365	20 h	<input checked="" type="checkbox"/> Reunión informativa sobre dimensión pacientes	
18.576	20 h	<input checked="" type="checkbox"/> Reunión informativa sobre dimensión profesionales	
18.688	4 h	<input checked="" type="checkbox"/> Cambiar zonas básicas excluidas en los calculos de todos los indicadores	
19.056	4 h	<input checked="" type="checkbox"/> Generación automática de perfiles v.2.39	
12.216	4 h	<input checked="" type="checkbox"/> Preamálisis	
13.475	12 h	<input checked="" type="checkbox"/> N_medicamentos	
13.318	4 h	<input checked="" type="checkbox"/> Cambiar indicador, % con plan terapéutico integral	
12.249	20 h	<input checked="" type="checkbox"/> Analizar nueva solución para datos de crónicos	
19.383	40 h	<input checked="" type="checkbox"/> 1. Crear ETL para anonimizar	
	2 h	<input checked="" type="checkbox"/> 2. Comunicar cambios a operacionales	
	24h	3. Adaptar paso 1 a la respuesta de los operacionales	se realizará en sprint4
17.414	8 h	<input checked="" type="checkbox"/> Evolucionar estratificador Poblacional por GMA	
13.865	4 h	<input checked="" type="checkbox"/> Crear base de datos Aux_Ssias_BI	
13.815	8 h	<input checked="" type="checkbox"/> Cambiar indicador número de pacientes polimedicados	
13.704	80 h	<input checked="" type="checkbox"/> Incorporar tabla Eunat en la BBD poblacional	
19.056		Nuevo modelo de indicadores	
	56h	<input checked="" type="checkbox"/> Análisis y diseño	

	40 h	<input checked="" type="checkbox"/>	Prueba de concepto	
	64 h	<input checked="" type="checkbox"/>	Desarrollo	

10.2.3 Sprint 4

SPRINT 4 06/04/2017 - 02/05/2017				
Id Tarea	Horas		Descripción	Aclaraciones
	120 h	<input checked="" type="checkbox"/>	Crear CdM AP en el nuevo modelo, indicadores	
	96 h	⚙	Crear nuevo modelo que permita agregar	(Realizaremos solo la parte de análisis 16h)
12.216 13.946	240 h	<input checked="" type="checkbox"/>	Copiar caja blanca 1of3 a la versión 3.x	
	40 h	<input checked="" type="checkbox"/>	Repasar USP e indicadores	
	48 h	<input checked="" type="checkbox"/>	Observa, mirando el metadato	
		<input checked="" type="checkbox"/>	Diseño	
		<input checked="" type="checkbox"/>	Implementar (según diseño)	
12.479	24 h	x	Añadir 1 indicador	
	32 h	<input checked="" type="checkbox"/>	despliegue 2.4	
	8 h	<input checked="" type="checkbox"/>	despliegue 2.39	
	8 h	<input checked="" type="checkbox"/>	despliegue 3.x	
	55 h	⚙	Documentación (DLyF y MC)	sin finalizar

10.2.4 Sprint 5

SPRINT 5 05/05/2017 - 23/05/2017				
Id Tarea	Horas		Descripción	Aclaraciones
19.523	240 h	<input checked="" type="checkbox"/>	Copiar caja blanca 2of3 a la versión 3.x	
19.056	40 h	⚙	MC	sin acabar
13.947	40 h	<input checked="" type="checkbox"/>	Indicadores (5nuevos) Alta / Execute	
11.830	16 h	<input checked="" type="checkbox"/>	Prueba de concepto	
			Tableau	
	16 h	<input checked="" type="checkbox"/>	Desktop (manual, catálogo producto)	
	16 h	<input checked="" type="checkbox"/>	Server	
	16 h	<input checked="" type="checkbox"/>	Seguridad	buscar opción para que no permita actualizar
13.312	24 h	<input checked="" type="checkbox"/>	Tareas de operación (añadir desinstalación de otras versiones
14.254	32 h	x	Revisar todos los indicadores	
13.313	40 h	x	GMA	
14.419	20 h	<input checked="" type="checkbox"/>	PdC Esquizofrenia, cambiar fuentes	
13.085		x	Cambiar campo Distribución_especciales	
18.576	8 h	<input checked="" type="checkbox"/>	Desplegar 3.1	

10.3 Procedimiento para el cálculo del indicador número de pacientes polimedicados

```

USE [IM_Cronicos]
GO
/***** Object: StoredProcedure [reportATENEA].[usp_ATENEA_AP_CGM_Carga_M_POLIMEDICADOS]    Script Date: 12/06/2017 12:58:54 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
--CREATE PROCEDURE
--=====
ALTER PROCEDURE [reportATENEA].[usp_ATENEA_AP_CGM_Carga_M_POLIMEDICADOS]
(
    @Fdt_fecha DATETIME = NULL
)
--=====
--
----MMSSE:      usp_ATENEA_AP_CGM_Carga_M_POLIMEDICADOS
----CREADO:      XXXX      2015.03.XX      Version Inicial
----ACTUALIZADO: XXXX      2015.05.19      Añadida fecha de referencia para el calculo del indicador. Es un parametro de entrada
----          XXXX      2015.07.28      Corregida la formula de calculo. Se restan pafales y otros gastos
----          XXXX      2015.07.29      Corregido el calculo de fecha de inicio. La ultima semana se corresponde a los ultimos 7 dias, el ultimo mes a las ultimas 4 semanas,
----                                     los ultimos 2 meses a las ultimas 8 semanas, el ultimo trimestre a los ultimos 3 meses, el ultimo semestre a los ultimos 6 meses y
----                                     el ultimo año al ultimo año. Las fechas obtenidas son inclusivas (para su uso en las queries)
----          XXXX      2015.08.20      Añadido historico en los indicadores de AP. En la practica hay ahora un nuevo atributo FE_Fecha en la tabla
----          XXXX      2016.04.04      Nuevo Universo de pacientes
----          XXXX      2016.04.22      Añadido fecha de alta como cronico en la clausula where
----          Dñue Ayechu      2017.03.16      Modificación del indicador: ahora se calcula el porcentaje de pacientes polimedificados del total de pacientes crónicos. (antes solo el numero de pacientes polimedificados)
----                                     No tener en cuenta los medicamentos como gases, pafales etc. (Los nuevos medicamentos a desestimar son: cod_tipo_producto: -1 y 14,
----                                     además de los ya desestimados anteriormente)
----          Dñue Ayechu      2017.03.30      Cambio especificaciones de negocio, anteriormente se contaban los medicamentos que se tomaban dentro de un periodo de 2 meses.
----                                     Actualmente, se quieren las medicaciones crónicas activas que llevan más de 2 meses de DURACION en el momento de sacar el dato
--
----DESCRIPCION:
--Calcula el porcentaje de pacientes polimedificados en los ultimos 2 meses, como denominador usamos la población diana. (actualiza la tabla ATENEA_AP_CUADRO_MANDO)
--EJEMPLO EJECUCION PROCEDIMIENTO: EXEC reportATENEA.usp_ATENEA_AP_CGM_Carga_M_POLIMEDICADOS
--DECLARE @dt_getdate AS DATETIME; SET @dt_getdate = GETDATE(); PRINT @dt_getdate; EXEC reportATENEA.usp_ATENEA_AP_CGM_Carga_M_POLIMEDICADOS @dt_getdate;
--PARAMETROS DE ENTRADA:
--@Fdt_fecha      Fecha de referencia para el calculo de los indicadores
--                  Si su valor no viene dado coje el valor de la tabla de configuración de KPI
--=====

AS
SET NOCOUNT ON
/* Return Values
0      Successfull execution of Package
<0    Package error
*/

--DECLARACION DE VARIABLES GENERICAS
--=====
--Stores error value (@@ERROR)
DECLARE @int_errorcode INT
SET @int_errorcode = 0

--Stores error description (@@ERROR)
DECLARE @vchr_errormsg VARCHAR(500)
SET @vchr_errormsg = NULL

--Proc execution return value
DECLARE @int_sperrocode INT
SET @int_sperrocode = 0

--Proc error return message
DECLARE @vchr_sperromsg VARCHAR(500)
SET @vchr_sperromsg = NULL

--Proc error message return data (typically an output parameter)
DECLARE @OP_VCHR_ERRORMSG VARCHAR(4000)

--Proc name
DECLARE @vchr_pname VARCHAR(25)
SET @vchr_pname = 'usp_ATENEA_AP_CGM_Carga_M_POLIMEDICADOS'

--DECLARACION DE VARIABLES DES PROCEDIMIENTO
--=====
--Stores SQL statements (to be executed)
DECLARE @vchr_sql VARCHAR(4000)
DECLARE @vchr_sql NVARCHAR(4000)

--Fecha final de ejecución. Sirve de base para el calculo de los indicadores. Es la fecha del parametro de entrada.
DECLARE @dt_fecharfin DATETIME

--Fechas iniciales de ejecución. Depende del indicador a calcular
DECLARE @dt_fecharini_semana DATETIME
DECLARE @dt_fecharini_mes DATETIME
DECLARE @dt_fecharini_2mes DATETIME
DECLARE @dt_fecharini_trimestre DATETIME
DECLARE @dt_fecharini_semestre DATETIME
DECLARE @dt_fecharini_año DATETIME

--=====
--CODIGO
--=====
--Valida parametros entrada
IF @Fdt_fecha IS NULL BEGIN
    --RAISERROR ('ERROR - Parametro fecha es nulo.', 16, 1)
    SELECT @dt_fecharfin = mst01.FECHA_CALCULO
    FROM [report].[KPI_CONFIG] mst01
    WHERE mst01.COD_INFORME = 'ATENEA_AP_CUADRO_MANDO'

    IF @dt_fecharfin = CONVERT(DATETIME, '99991231', 112) BEGIN
        SET @vchr_sql = CONVERT(VARCHAR, @dt_fecharfin, 112)
        RAISERROR ('ERROR - Parametro fecha invalido. Valor por defecto en la tabla KPI_CONFIG es %s', 16, 1, @vchr_sql)
    END
END
ELSE BEGIN
    SET @dt_fecharfin = @Fdt_fecha
END

--Gestiona registro de configuración de ATENEA_AP_CUADRO_MANDO
--Si la fecha de referencia para el calculo es distinta a la establecida en la tabla de configuración se pierde la coherencia en los datos presentados (unos tendran fecha de ref diferente)
IF (SELECT mst01.FECHA_CALCULO FROM [report].[KPI_CONFIG] mst01 WHERE mst01.COD_INFORME = 'ATENEA_AP_CUADRO_MANDO') <> @dt_fecharfin BEGIN
    UPDATE [report].[KPI_CONFIG]
    SET COD_COHERENCIA = 0,
        FECHA_CARGAMODIF = GETDATE(),
        USUARIO_CARGAMODIF = CURRENT_USER
    WHERE COD_INFORME = 'ATENEA_AP_CUADRO_MANDO'
END
ELSE BEGIN
    UPDATE [report].[KPI_CONFIG]
    SET FECHA_CARGAMODIF = GETDATE(),
        USUARIO_CARGAMODIF = CURRENT_USER
    WHERE COD_INFORME = 'ATENEA_AP_CUADRO_MANDO'
END
END

```

```

SET @int_errorcode = @@ERROR
IF @int_errorcode <> 0 OR @int_sperrocode <> 0 BEGIN
    GOTO ERROR_HANDLER
END

--Resetea indicador
DELETE FROM [reportATENEA].[ATENEA_AP_CUADRO_MANDO]
WHERE FK_INDICADOR = 46
AND FK_FECHA = CONVERT(INT, CONVERT(VARCHAR, @dt_fechafin, 112))

SET @int_errorcode = @@ERROR
IF @int_errorcode <> 0 OR @int_sperrocode <> 0 BEGIN
    GOTO ERROR_HANDLER
END

--Calcula fecha inicial (ultimos 2 meses)
SET @dt_fechaini_2mes = DATEADD(DAY, 1, DATEADD(WEEK, -8, @dt_fechafin))

SET @int_errorcode = @@ERROR
IF @int_errorcode <> 0 OR @int_sperrocode <> 0 BEGIN
    GOTO ERROR_HANDLER
END

--Calcula indicador
INSERT INTO [reportATENEA].[ATENEA_AP_CUADRO_MANDO]
(
    FK_FECHA
    ,FK_INDICADOR
    ,SK_PLAZAS_AP
    ,VALOR
    ,VALOR_ZONA
    ,VALOR_GLOBAL
    ,FECHA_CARGA
    ,Perfil_Paciente
)
SELECT CONVERT(INT, CONVERT(VARCHAR, @dt_fechafin, 112)) FK_FECHA
,46 FK_INDICADOR
,A.SK_PLAZAS_AP
,A.VALOR_PLAZA
,A.VALOR_ZONA
,A.VALOR_GLOBAL
,GETDATE() FECHA_CARGA
,A.PERFIL_PACIENTE
FROM (
    SELECT COALESCE(PA.SK_PLAZAS_AP, '-1') SK_PLAZAS_AP
    ,T.perfil_PACIENTE
    , COALESCE(SUM(CASE WHEN T.sk_Paciente_Full IS NOT NULL THEN 1 ELSE 0 END) OVER(PARTITION BY COALESCE(PA.SK_PLAZAS_AP, '-1')) / NULLIF(CAST(COUNT(T.SK_PACIENTE) OVER(PARTITION BY COALESCE(PA.SK_PLAZAS_AP, '-1')) AS float), 0), 0) VALOR_PLAZA
    , COALESCE(SUM(CASE WHEN T.sk_Paciente_Full IS NOT NULL THEN 1 ELSE 0 END) OVER(PARTITION BY COALESCE(PA.COD_ZONA_BASICA, '-1')) / NULLIF(CAST(COUNT(T.SK_PACIENTE) OVER(PARTITION BY COALESCE(PA.COD_ZONA_BASICA, '-1')) AS float), 0), 0) VALOR_ZONA
    , COALESCE(SUM(CASE WHEN T.sk_Paciente_Full IS NOT NULL THEN 1 ELSE 0 END) OVER() / NULLIF(CAST(COUNT(T.SK_PACIENTE) OVER() AS float), 0), 0) VALOR_GLOBAL
    FROM (
        SELECT DISTINCT PE.SK_PACIENTE, PE.CIAS, B.sk_Paciente as sk_Paciente_Full
        FROM [dbo].[DIM_PACIENTES_EXT] PE
        LEFT JOIN [InsusioServarra].[DIM_PACIENTE_CRONICO_PERFIL] Perfil ON PE.SK_PACIENTE = Perfil.SK_PACIENTE
        LEFT JOIN (
            SELECT count(DISTINCT del02.COD_GRUPO_TERAREUTICO) N_MEDICAMENTOS, sk_Paciente sk_Paciente
            FROM [Iamia].[ITH_PRESCRIPCIONES_DISPENSACIONES] met01
            INNER JOIN [Iamia].[DIM_TIPO_PRESCRIPCION] del01
            ON met01.SK_TIPO_PRESCRIPCION = del01.SK_TIPO_PRESCRIPCION
            INNER JOIN [Iamia].[DIM_MEDICAMENTOS] del02
            ON met01.SK_MEDICAMENTO_DISPENSADO = del02.SK_MEDICAMENTO
            WHERE del01.DES_TIPO_PRESCRIPCION = 'Circuito'
            AND (met01.FECHA_FIN_PRESCRIPCION > CONVERT(INT, CONVERT(VARCHAR, @dt_fechafin, 112))
                OR met01.fecha_fin_prescripcion = 18991230
                OR met01.fecha_fin_prescripcion is null
            )
            AND (DATEDIFF(week, CONVERT(DATETIME, CONVERT(CHAR, met01.FECHA_INICIO_PRESCRIPCION), 0), @dt_fechafin)) > 0
            AND del02.cod_tipo_producto NOT IN (-1, 7, 8, 14)
            AND (
                del02.cod_tipo_abouchente NOT IN (1, 2, 3)
                OR del02.cod_tipo_abouchente IS NULL
            )
        )
        GROUP BY SK_PACIENTE
        HAVING count(DISTINCT del02.COD_GRUPO_TERAREUTICO) >= 2
    ) B ON PE.SK_PACIENTE=B.sk_Paciente
    -- INICIO DEMONSTRADOR PACIENTES A FILTRAR
    WHERE PE.FECHA_PALCICHIENTO > @dt_fechafin
    AND (
        (PE.COD_ES_CRONICO = 1 AND PE.FECHA_ALTA_ES_CRONICO <= @dt_fechafin AND PE.COD_CIRCUITOS_ESPECIFICOS = '1')
        OR
        (Perfil.Descripcion_Perfil IN ('Fisiopatologias', 'Demencia', 'Insuf. Cardiaca', 'Diabetes', 'Fibromialgia')) -- EXCEP SEGUIMIENTO
    )
    OR
    (PE.COD_ES_CRONICO = 1 AND PE.FECHA_ALTA_ES_CRONICO <= @dt_fechafin AND DES_VALORACTOR_GLOBAL IN('Severo', 'Dilatorio'))
    )
    -- FIN DEMONSTRADOR PACIENTES A FILTRAR
    --Exclusion END
    AND PE.COD_ZONA_BASICA not in (select COD_ZONA_BASICA from report.DIM_Zonas_Excluidas)
    ) T FULL OUTER JOIN [dbo].[DIM_PLAZAS_AP] PA ON PA.CIAS = T.CIAS AND pa.SK_PLAZAS_AP < 0
) A
GROUP BY A.SK_PLAZAS_AP, A.PERFIL_PACIENTE, A.VALOR_PLAZA, A.VALOR_ZONA, A.VALOR_GLOBAL

SET @int_errorcode = @@ERROR
IF @int_errorcode <> 0 OR @int_sperrocode <> 0 BEGIN
    GOTO ERROR_HANDLER
END

-----
--TRATAMIENTO DE ERRORES GENERICO
-----

ERROR_HANDLER:
--ERRORS (step:[900-999])
-----
--system errors
IF @int_errorcode <> 0 BEGIN
    SELECT @vchr_errormsg = description
    FROM master.dbo.sysmessages
    WHERE error = @int_errorcode

    SET @OP_VCHR_ERRORMSG = 'Error(' + @vchr_pname + '): ' +
        CONVERT(VARCHAR, @int_errorcode) +
        ' ' + @vchr_errormsg

    RAISERROR (@OP_VCHR_ERRORMSG, 16, 1)
    RETURN @int_errorcode
END

--called proc errors
IF @int_sperrocode <> 0 BEGIN
    SET @OP_VCHR_ERRORMSG = 'Error(' + @vchr_pname + '): ' +
        ' ' + @vchr_sperromsg

    RAISERROR (@OP_VCHR_ERRORMSG, 16, 1)
    RETURN @int_sperrocode
END

-----
--CIERRE PROCEDIMIENTO
-----

RETURN @@ERROR

```